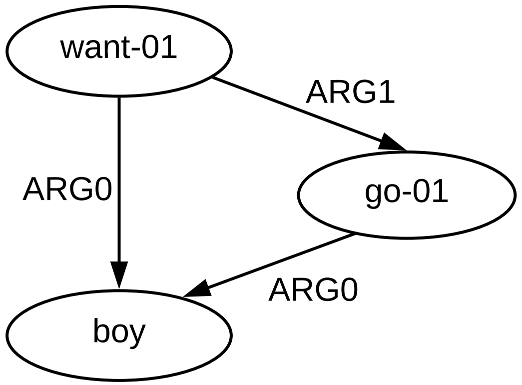
Tackling Graphical NLP problems with Graph Recurrent Networks (GRN)

Linfeng Song

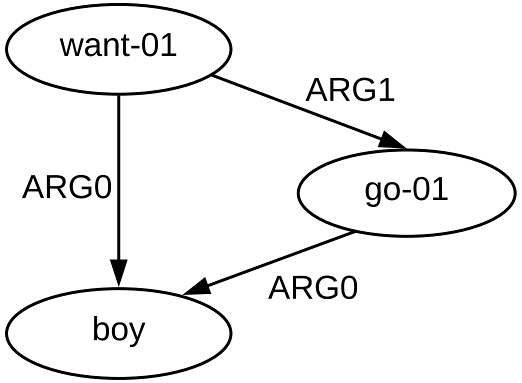
University of Rochester

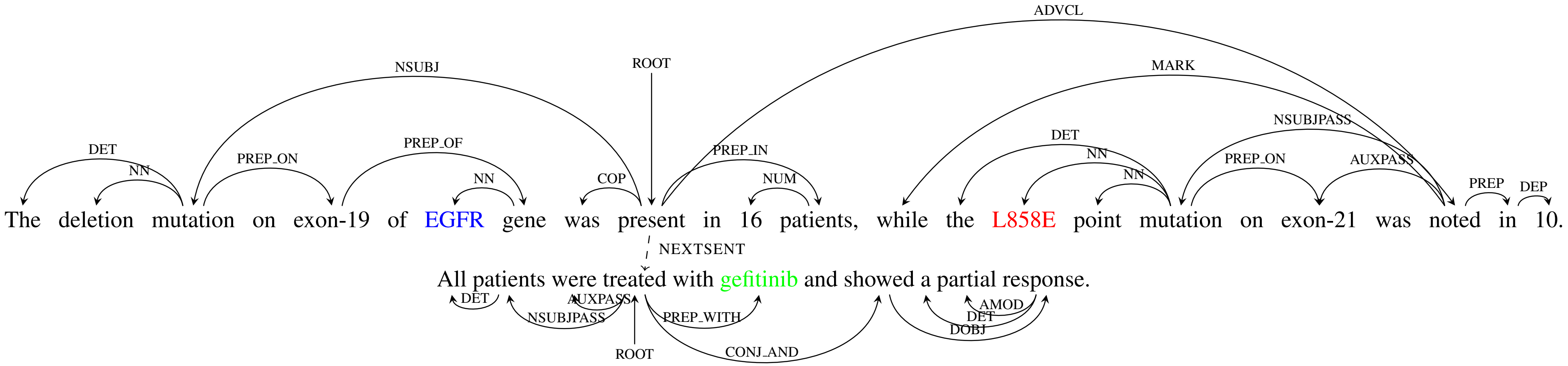
advised by professor Daniel Gildea

Graphical problems in NLP

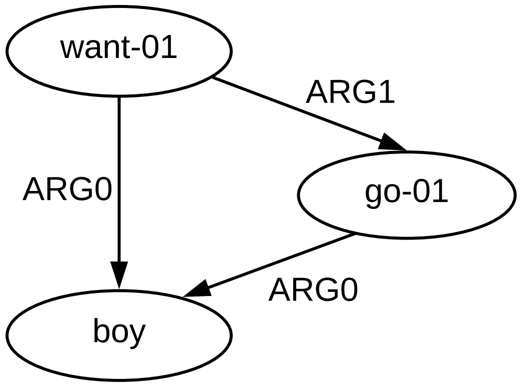


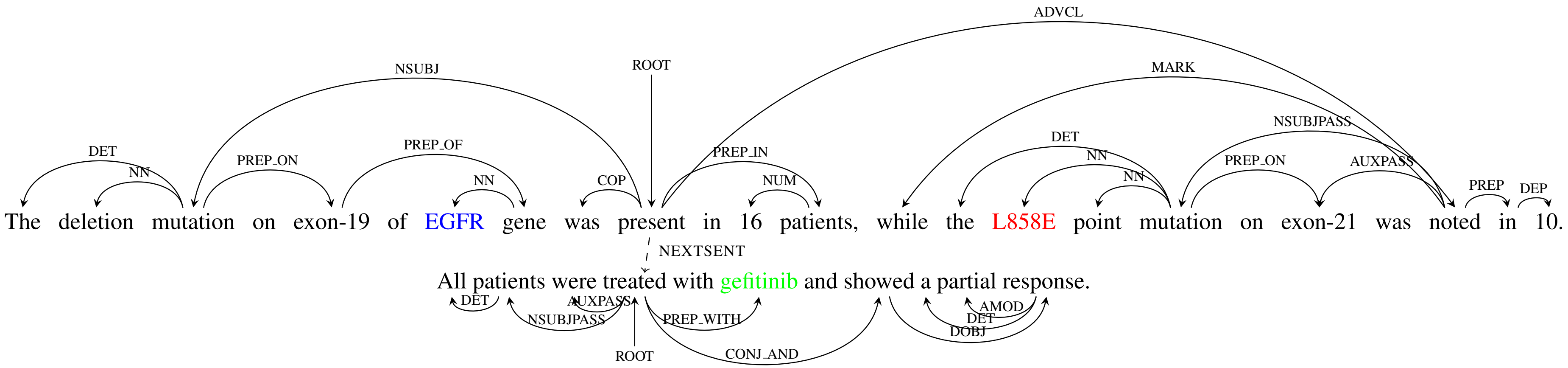
Graphical problems in NLP

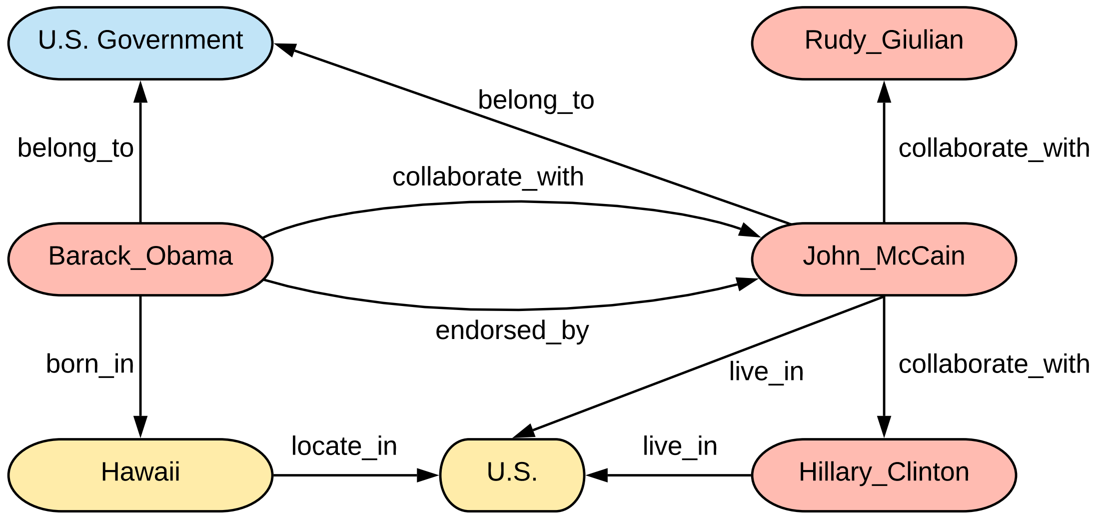




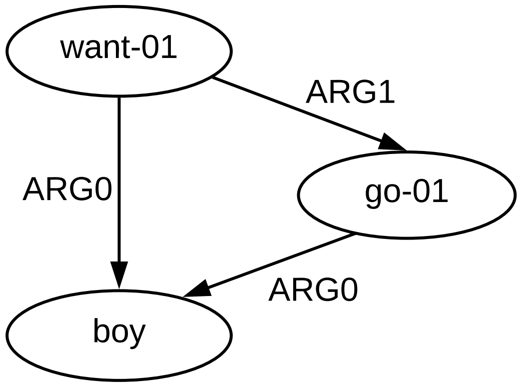
Graphical problems in NLP

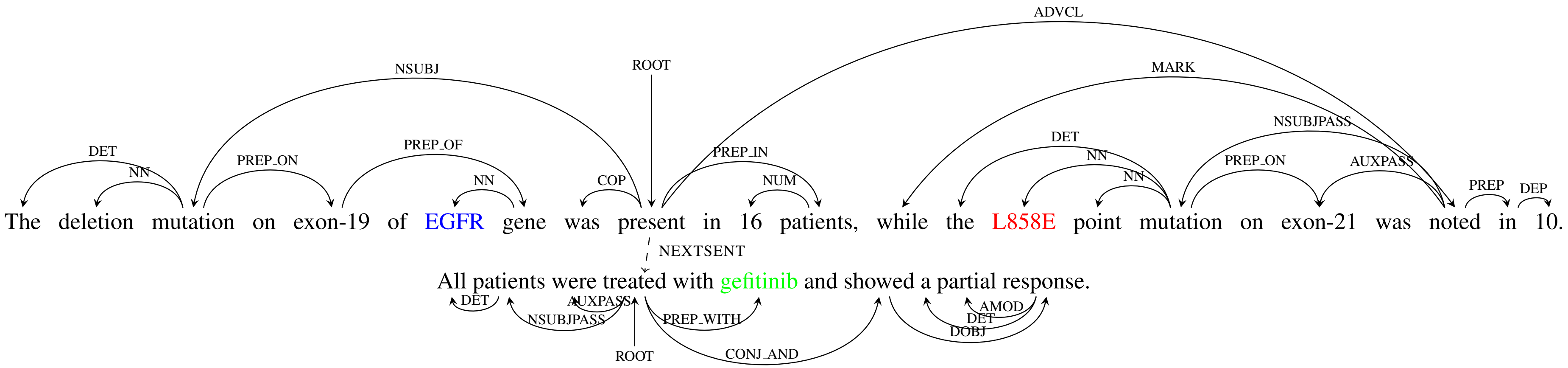


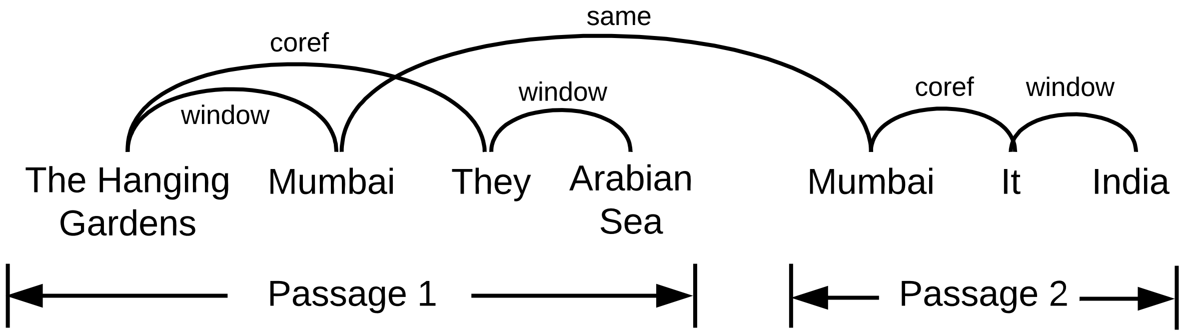


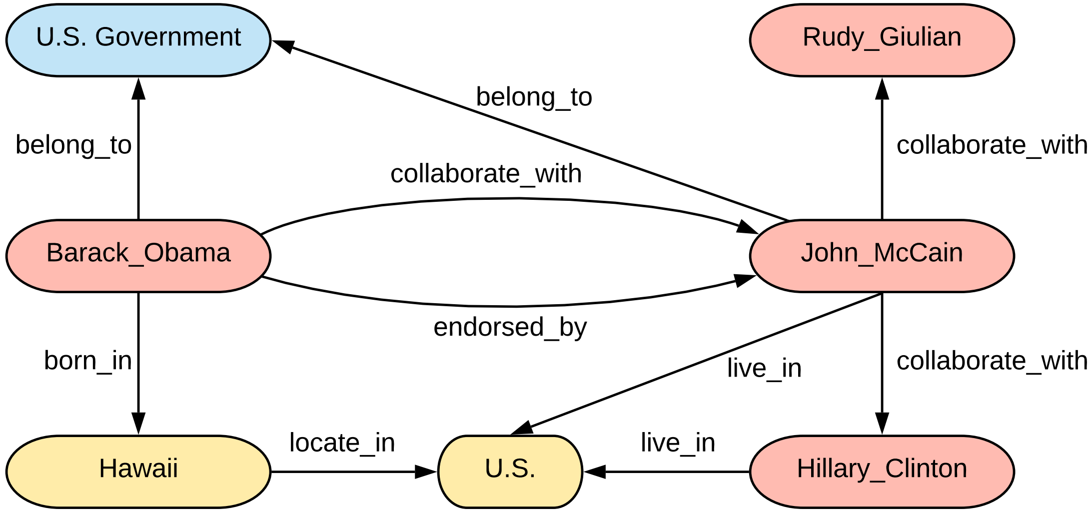


Graphical problems in NLP

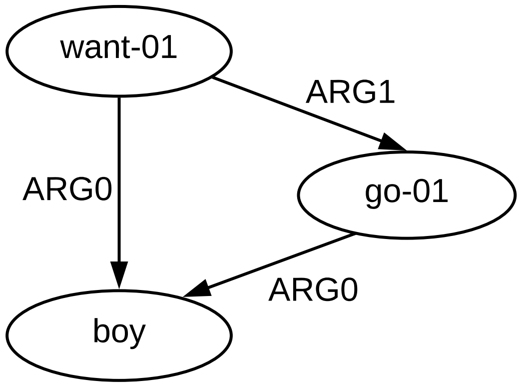


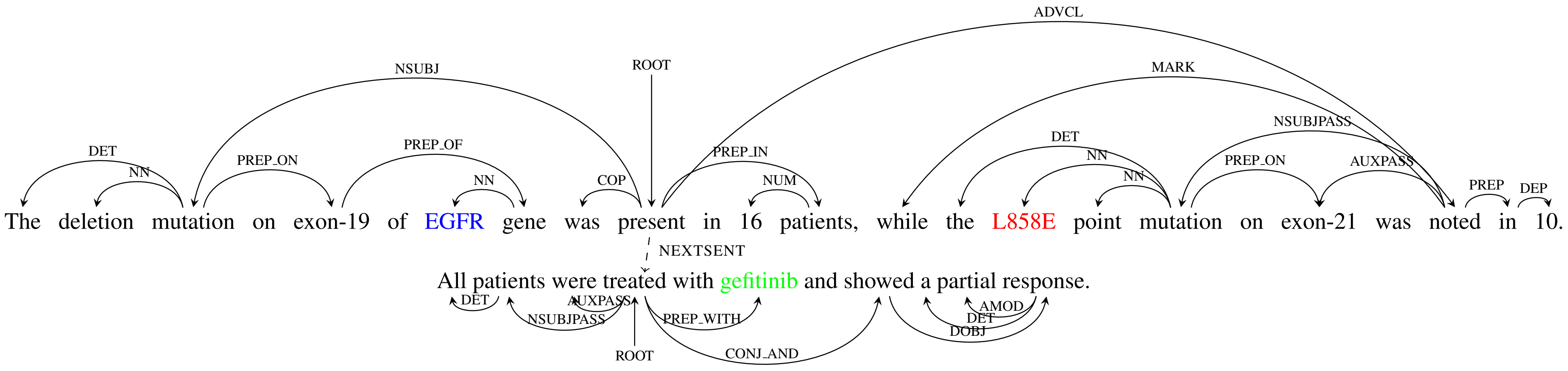


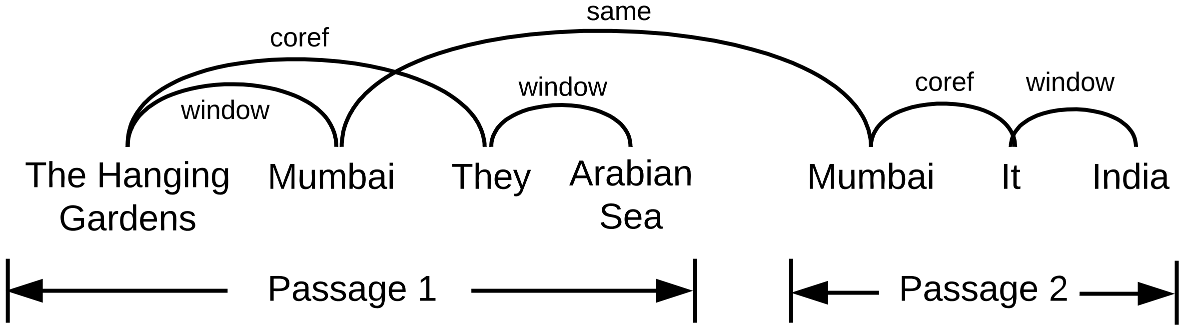


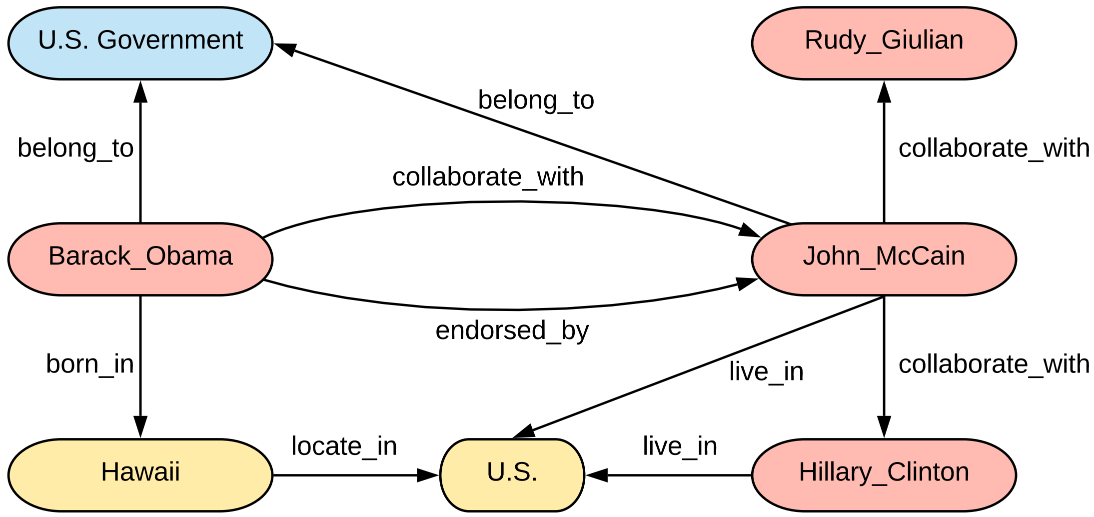


Graphical problems in NLP



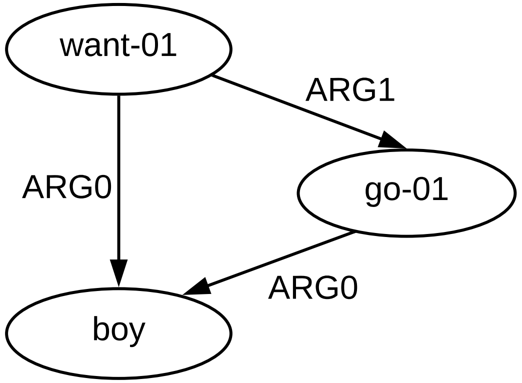


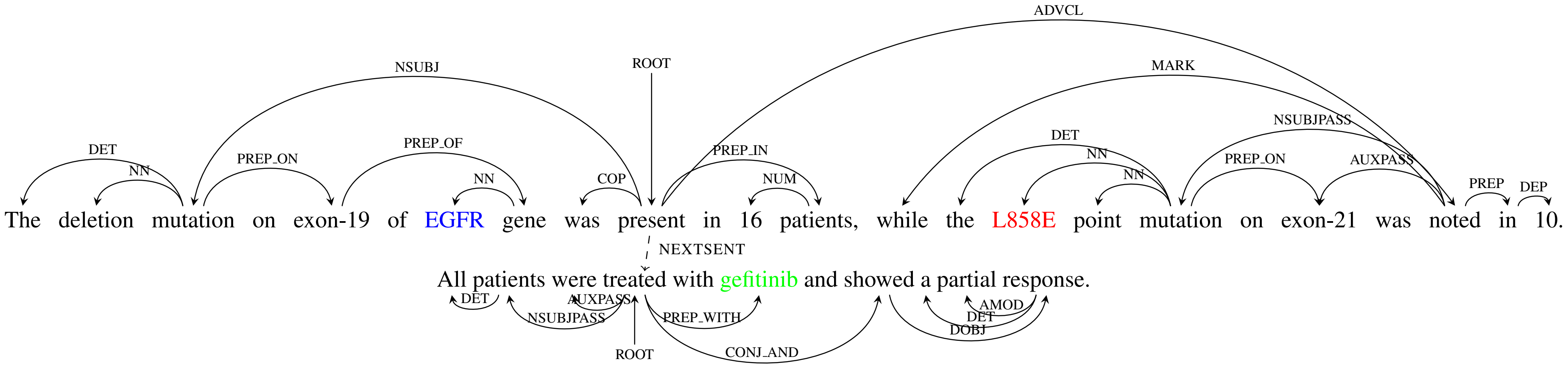


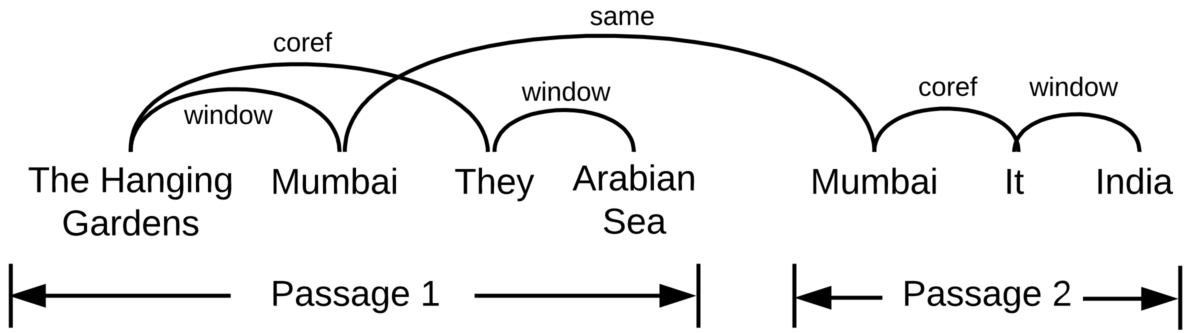


I $\leftrightarrow$ love $\leftrightarrow$ NLP

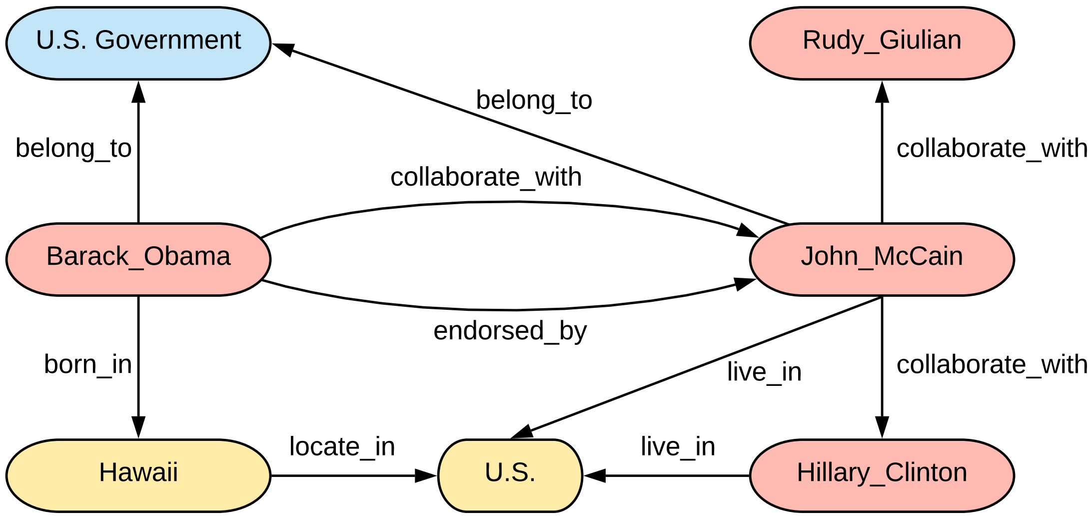
Graphical problems in NLP



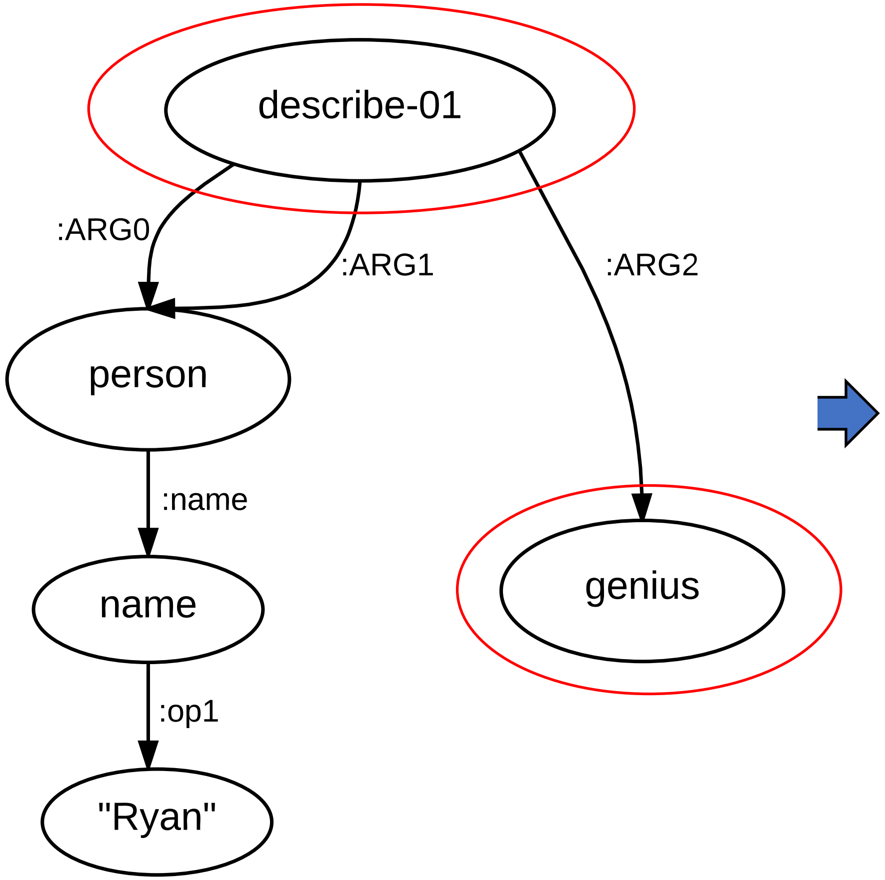




I ↔ love ↔ NLP . . .

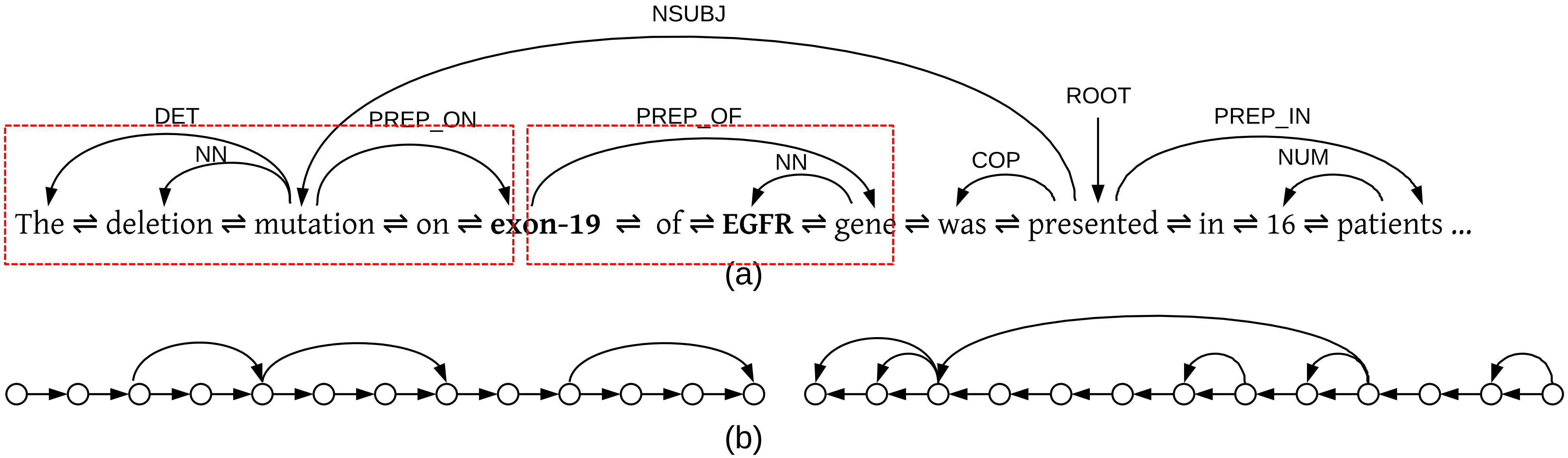


Previous work: linearization + RNN (Konstas et al., ACL 2017)

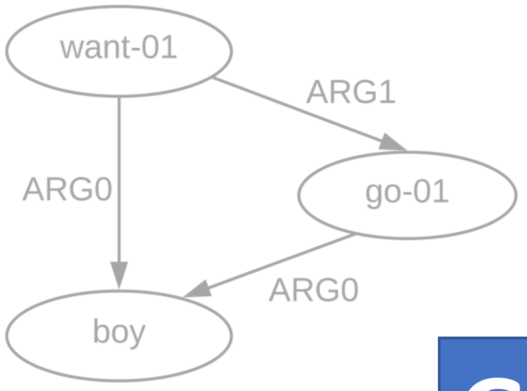


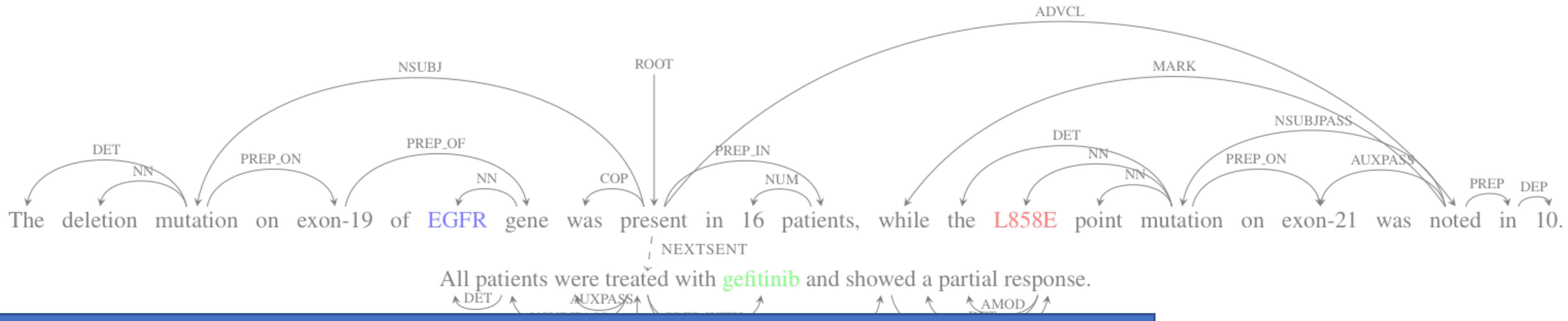
describe :arg0 (person :name (name :op1 Ryan)) :arg1 person :arg2 **genius**

Previous work: graph separation + DAG LSTM (Peng et al., TACL 2017)

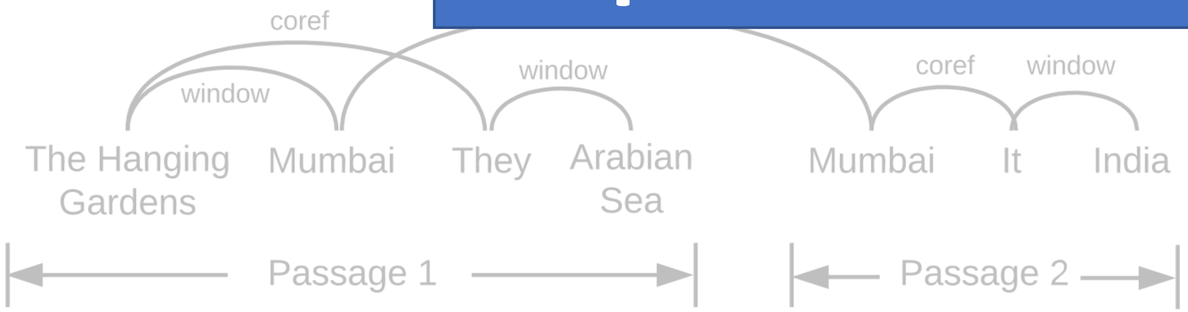


Graphical problems in NLP

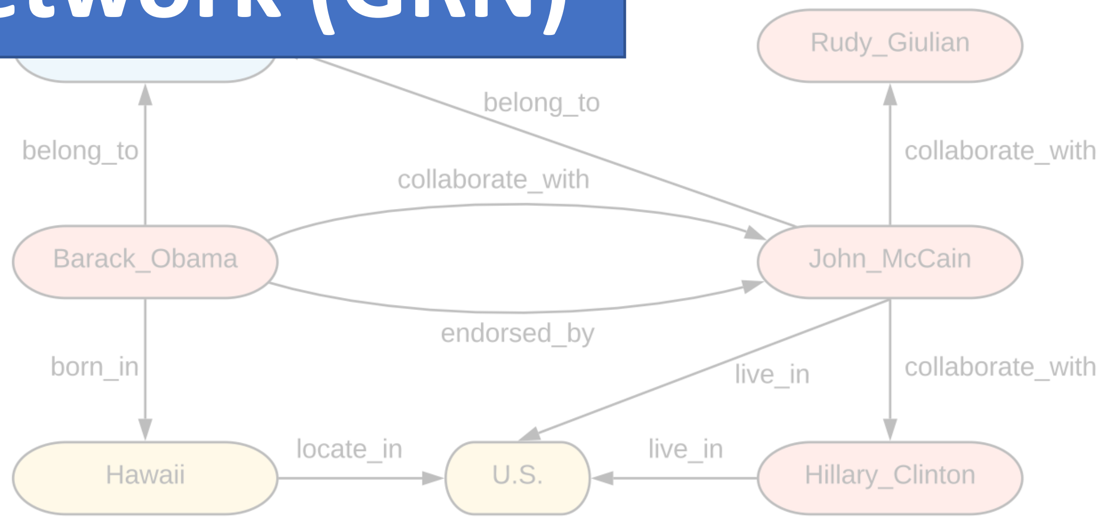




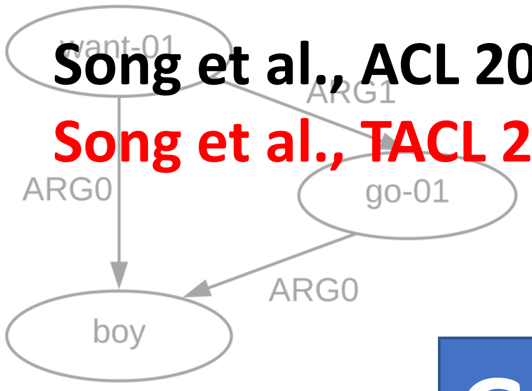
Graph recurrent network (GRN)



I ↔ love ↔ NLP

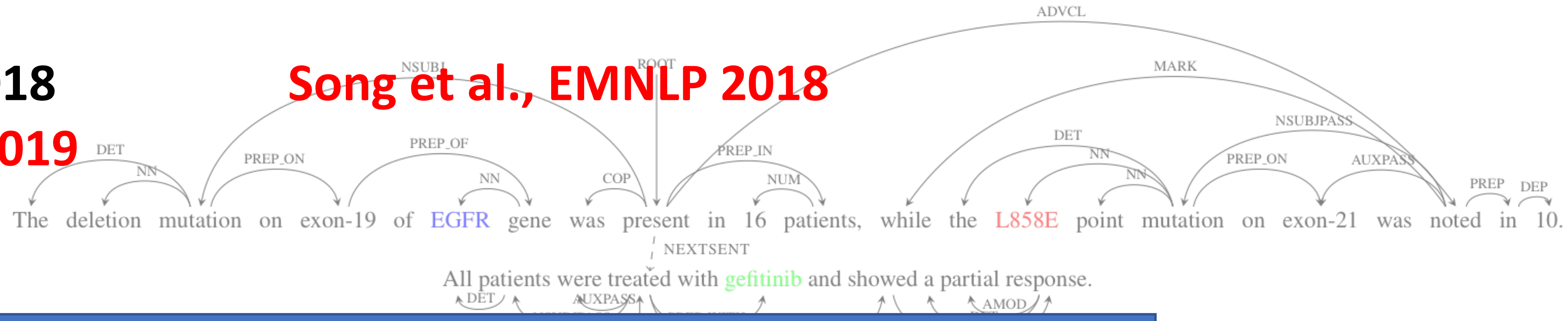


Graphical problems in NLP



Song et al., ACL 2018

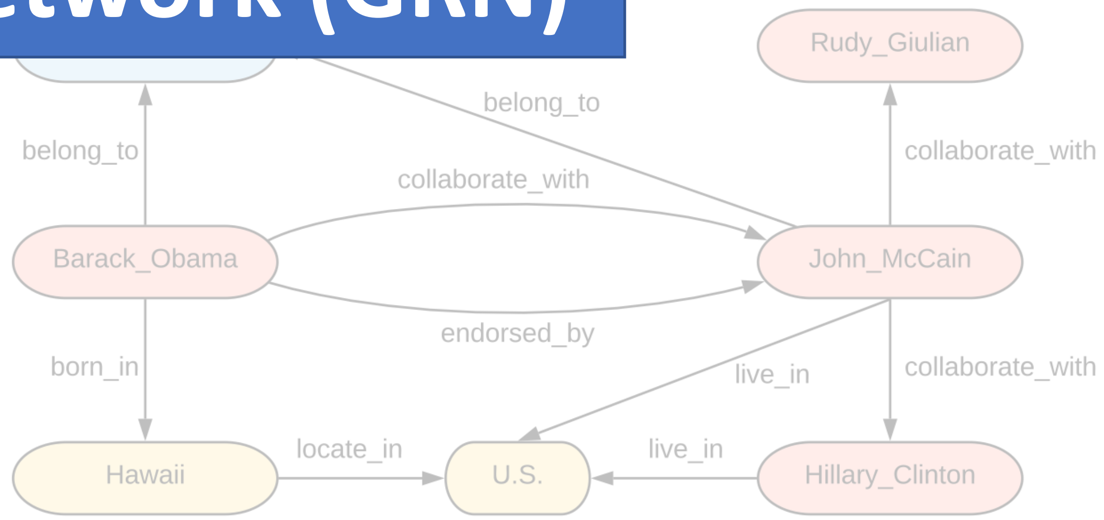
Song et al., TACL 2019



Song et al., EMNLP 2018

Graph recurrent network (GRN)

Song et al., Arxiv preprint 2018



I ↔ love ↔ NLP **Zhang et al., ACL 2018**

Outline

- Evidence Integration for Multi-hop Reading Comprehension with Graph Neural Networks.
- N-ary Relation Extraction using Graph State LSTM.
- Semantic Neural Machine Translation using AMR.

Multi-hop reading comprehension

Q: (The Hanging Gardens, country, ?)
Candidates: {Iran, India, Pakistan, Somalia, ...}

The Hanging Gardens, in [**Mumbai**], also known as Pherozeshah Mehta Gardens, are terraced gardens ... [**They**] provide sunset views over [**the Arabian Sea**] ...

[**Mumbai**] (also known as Bombay, the official name until 1995) is the capital city of the Indian state of Maharashtra. [**It**] is the most populous city in [**India**] ...

[**The Arabian Sea**] is a region of the northern Indian Ocean bounded on the north by [**Pakistan**] and [**Iran**], on the west by northeastern [**Somalia**] and the Arabian Peninsula ...

Multi-hop reading comprehension

Q: (The Hanging Gardens, country, ?)
Candidates: {Iran, India, Pakistan, Somalia, ...}

The Hanging Gardens, in [**Mumbai**], also known as Pherozeshah Mehta Gardens, are terraced gardens ... [**They**] provide sunset views over [**the Arabian Sea**] ...

[**Mumbai**] (also known as Bombay, the official name until 1995) is the capital city of the Indian state of Maharashtra. [**It**] is the most populous city in [**India**] ...

[**The Arabian Sea**] is a region of the northern Indian Ocean bounded on the north by [**Pakistan**] and [**Iran**], on the west by northeastern [**Somalia**] and the Arabian Peninsula ...

Relevant evidence:
- The Hanging Gardens are in Mumbai.
- Mumbai is the most populous city in India.

Irrelevant evidence:
- The Hanging Gardens provide sunset views over the Arabian Sea.
- The Arabian Sea is bounded by Pakistan, Iran and Somalia.

Multi-hop reading comprehension

Q: (The Hanging Gardens, country, ?)
Candidates: {Iran, India, Pakistan, Somalia, ...}

The Hanging Gardens, in [**Mumbai**], also known as Pherozeshah Mehta Gardens, are terraced gardens ... [**They**] provide sunset views over [**the Arabian Sea**] ...

(1) Structure creation

[**Mumbai**] (also known as Bombay, the official name until 1995) is the capital city of the Indian state of Maharashtra. [**It**] is the most populous city in [**India**] ...

(2) Evidence integration

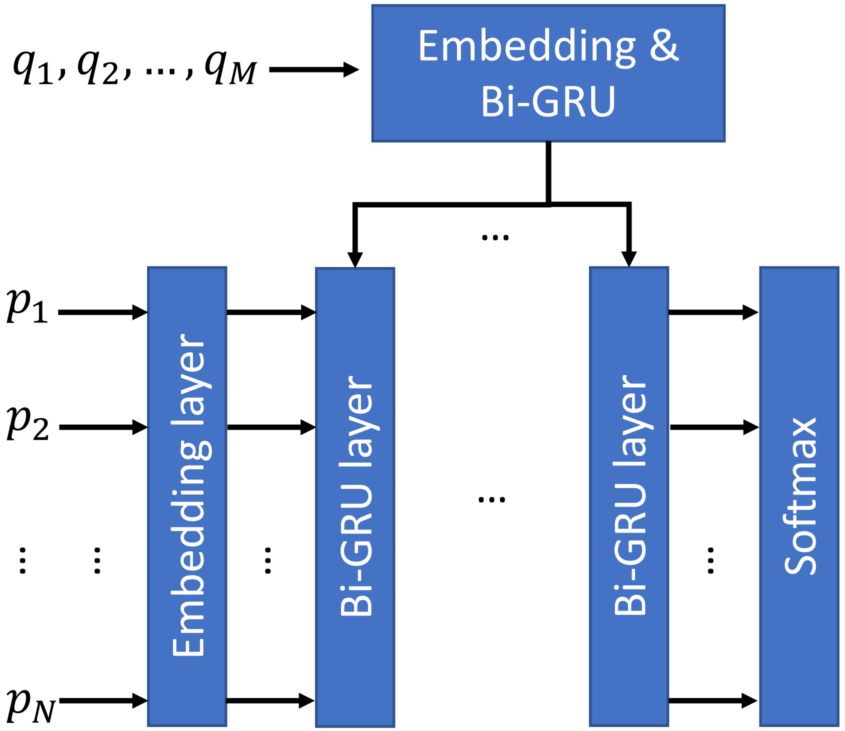
[**The Arabian Sea**] is a region of the northern Indian Ocean bounded on the north by [**Pakistan**] and [**Iran**], on the west by northeastern [**Somalia**] and the Arabian Peninsula ...

Relevant evidence:
- The Hanging Gardens are in Mumbai.
- Mumbai is the most populous city in India.

Irrelevant evidence:
- The Hanging Gardens provide sunset views over the Arabian Sea.
- The Arabian Sea is bounded by Pakistan, Iran and Somalia.

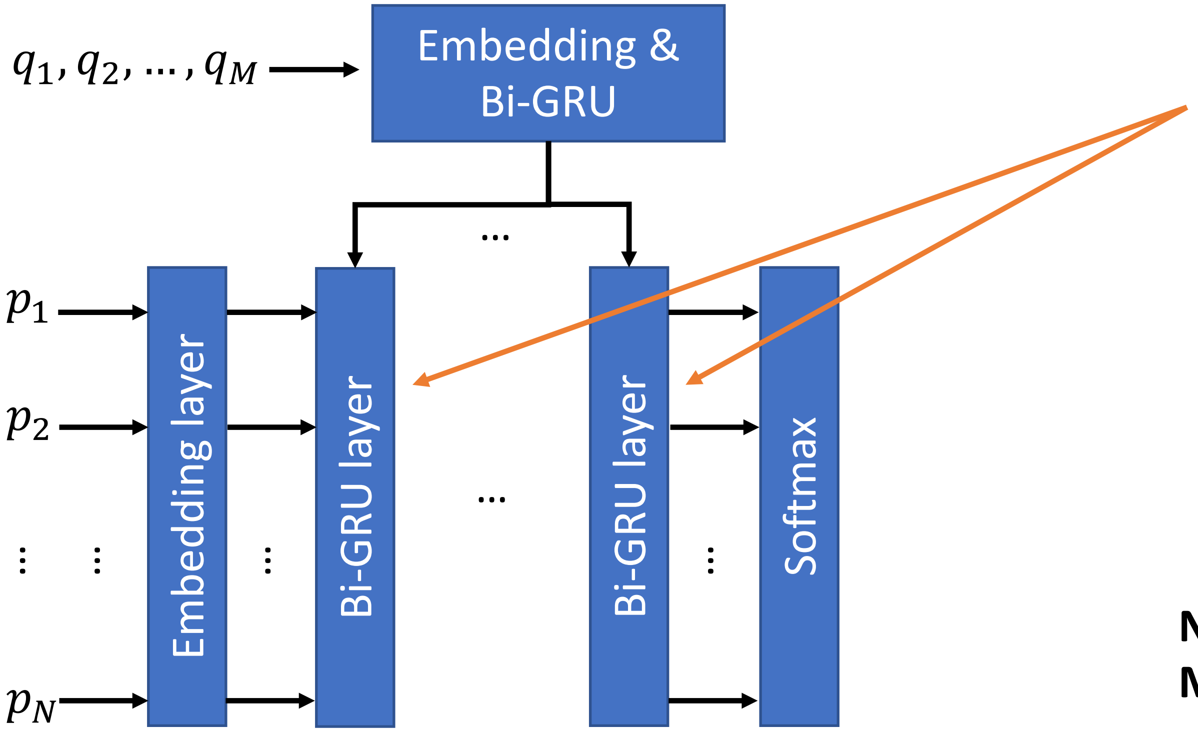
Previous SOTA (Dhingra et al., NAACL 2018)

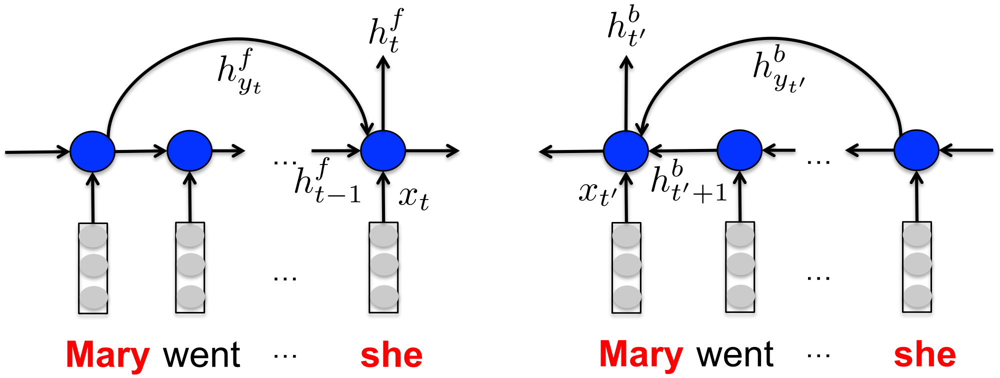
- Baseline: gated-attention reader (Dhingra et al., ACL 2017)



Designed for cloze-style reading comprehension

Previous SOTA (Dhingra et al., NAACL 2018)

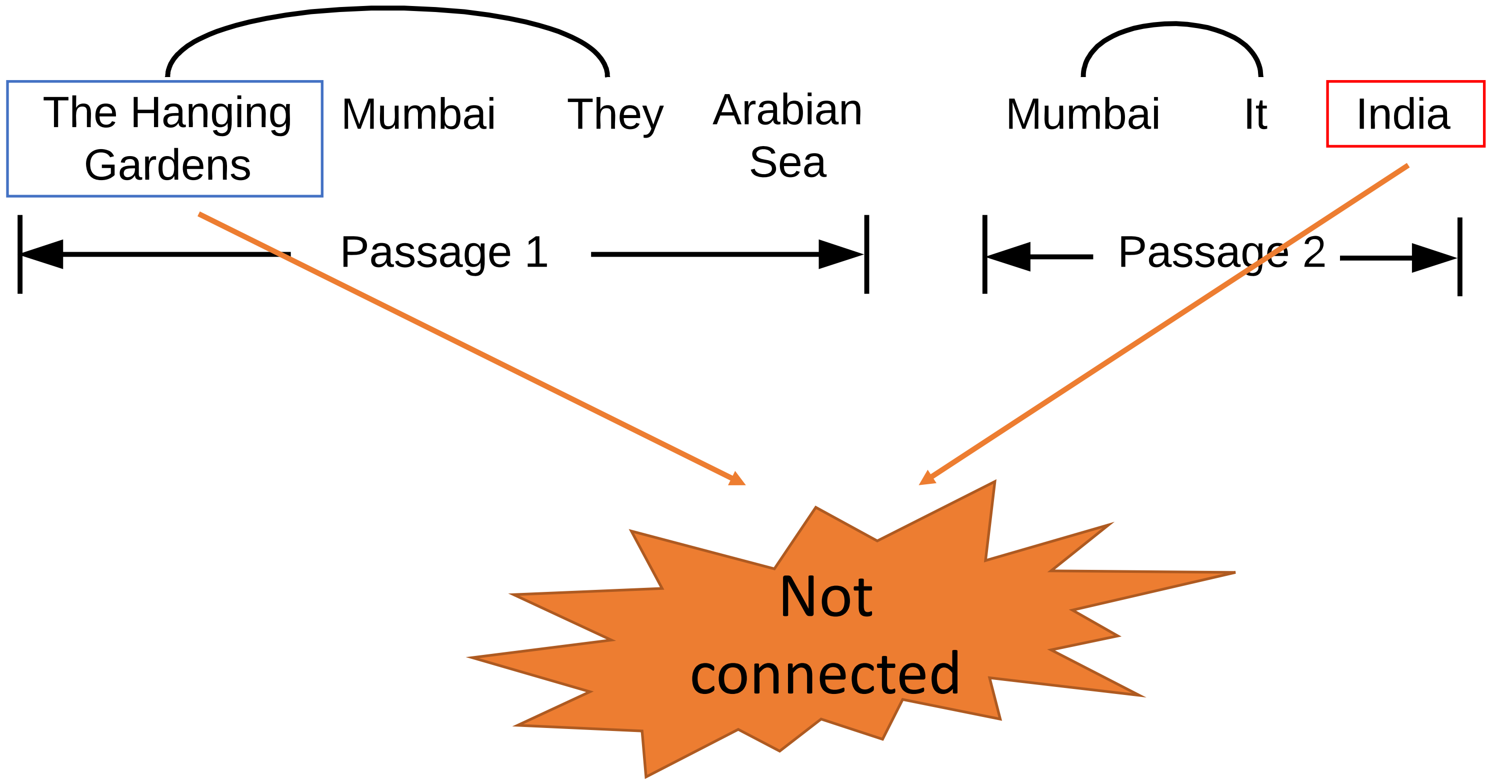




A coref-GRU layer

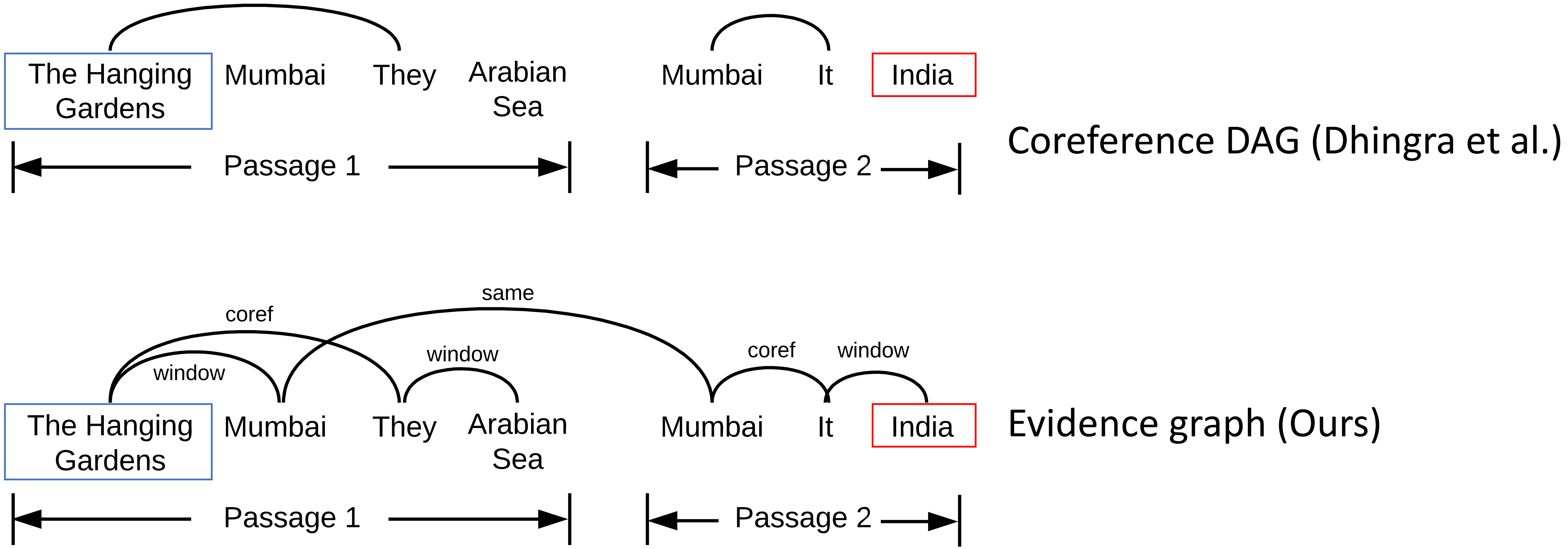
Neural Models for Reasoning over Multiple Mentions using Coreference (Dhingra et al., NAACL 2018)

Coref-DAG

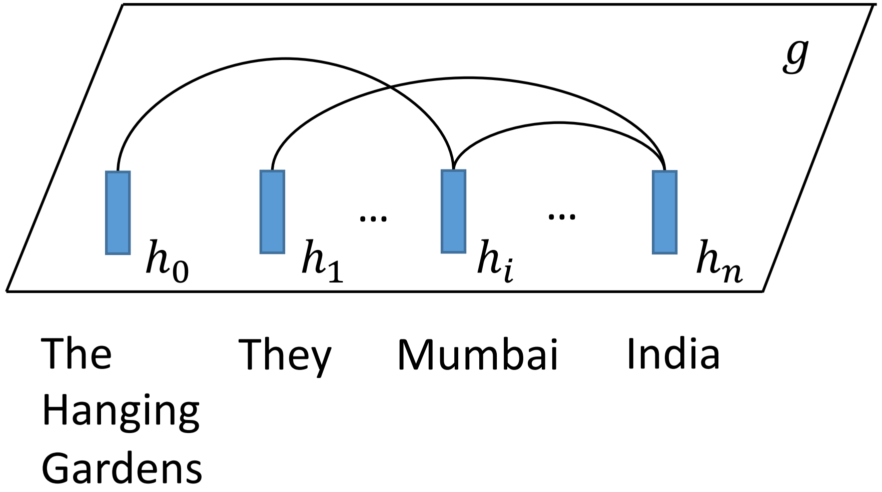


Coreference DAG (Dhingra et al.)

Coref-DAG vs Evidence graph

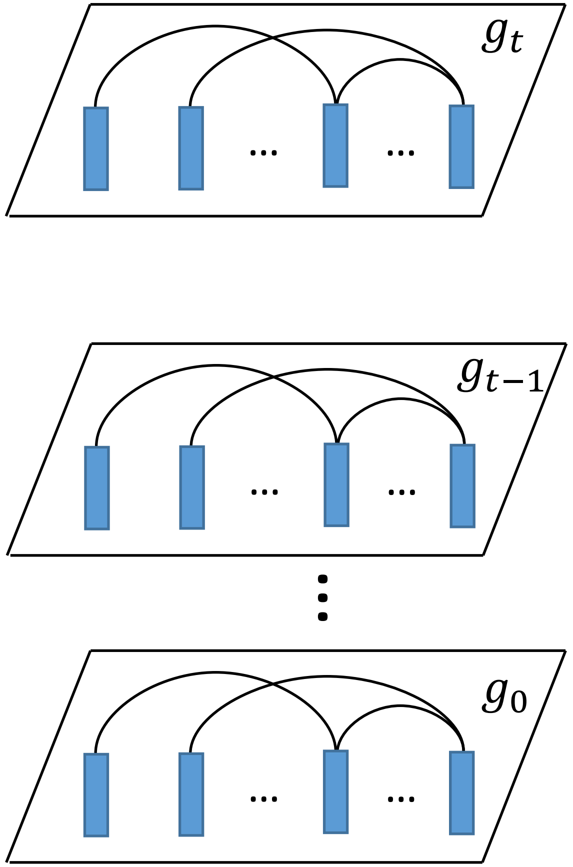


Graph recurrent network (GRN)



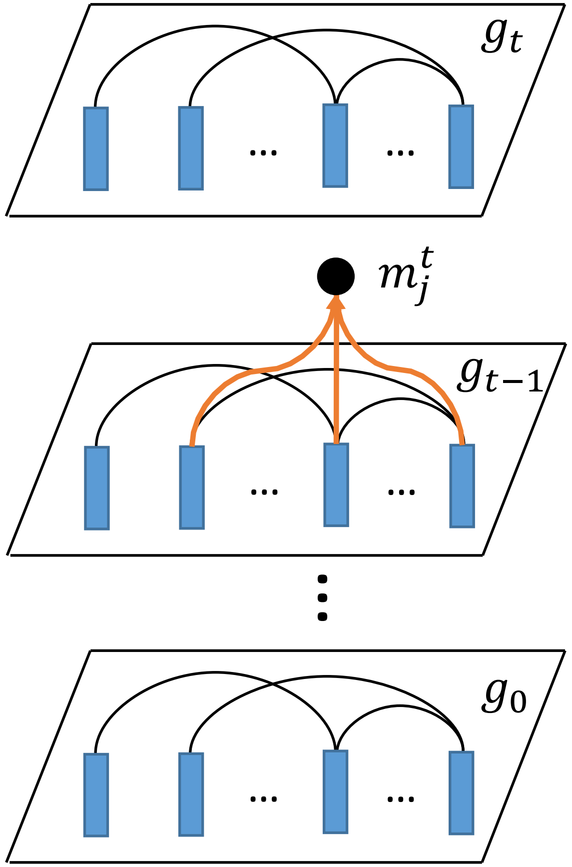
$$g = \{h_0, h_1 \ldots, h_n\}$$

Graph recurrent network (GRN)



- GRN follows an iterative message passing process for updating each node state. Within each iteration, it takes two main steps:
  - Message calculation
  - Node state update

Graph recurrent network (GRN)

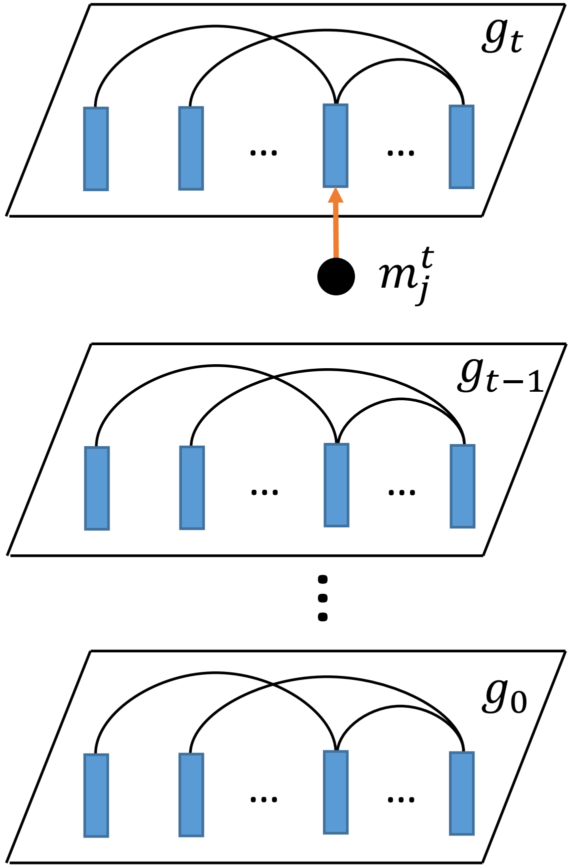


- Messages are first calculated by summing up the hidden states of neighbors

$$m_j^t = \sum_{i \in N_j} h_i^{t-1}$$

N_j: all neighbors of v_j

Graph recurrent network (GRN)



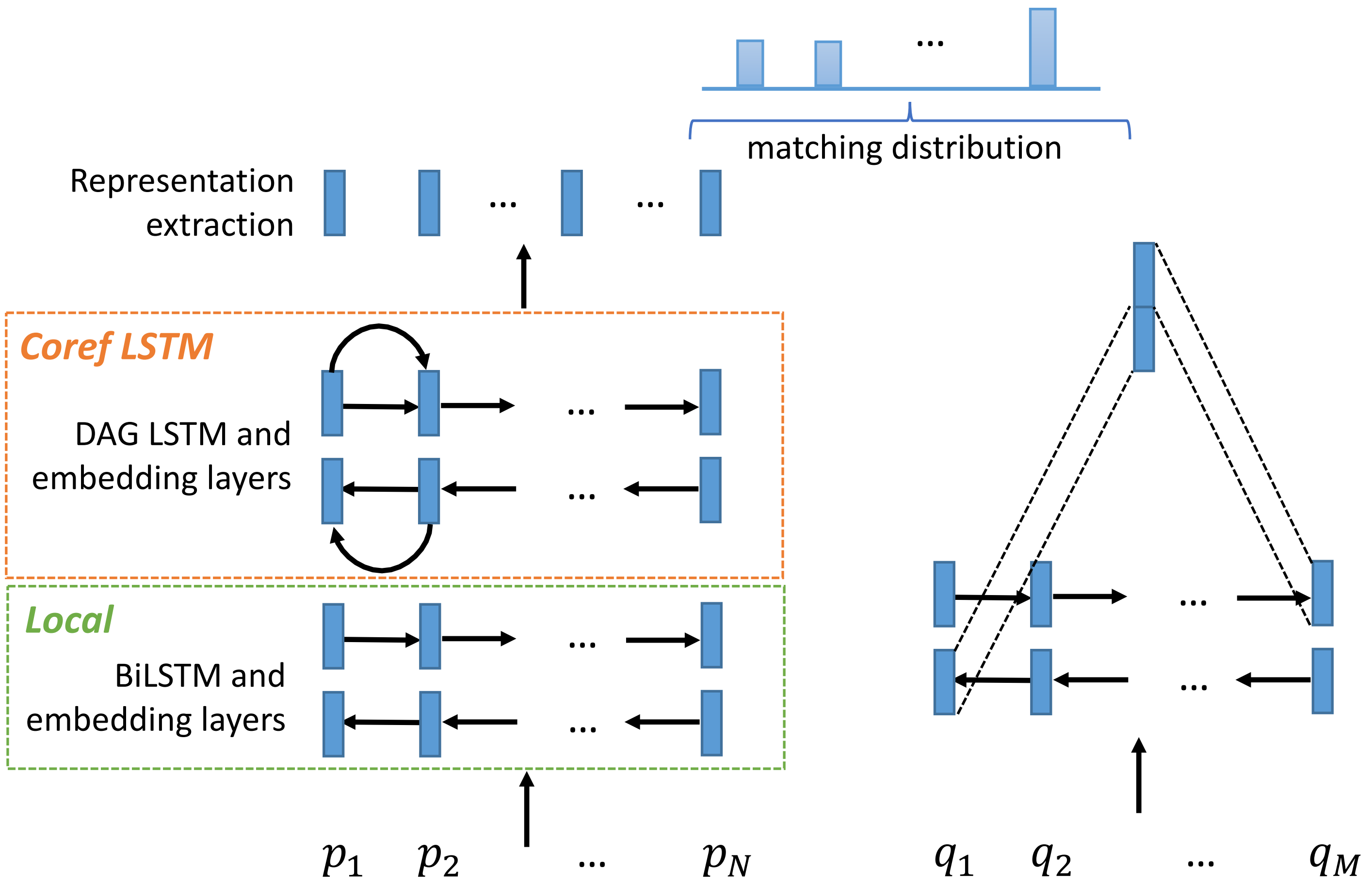
- Node states are updated with messages through an LSTM step.

$$h_j^t, c_j^t = LSTM(m_j^t, c_j^{t-1})$$

Comparing GRN with other GNNs

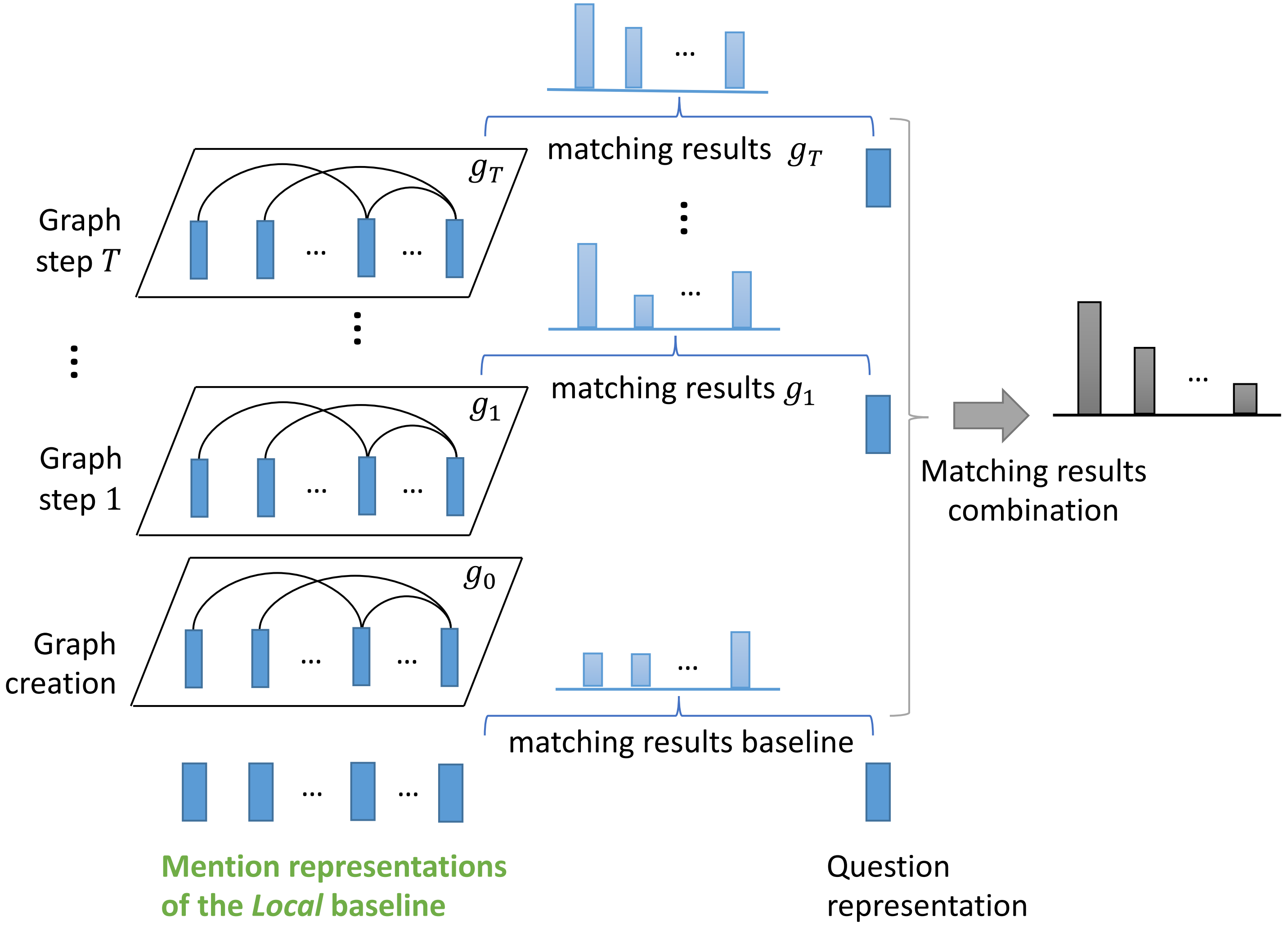
	GRN (ACL 2018)	GCN (EMNLP 2017)	GGNN (ACL 2018)
Message calculation:	$m_j^t = \sum_{i \in N_j} h_i^{t-1}$		
State update:	$h_j^t, c_j^t = LSTM(m_j^t, [h_j^{t-1} c_j^{t-1}])$	$h_j^t = \sigma(W m_j^t + b)$	$h_j^t = GRU(m_j^t, h_j^{t-1})$
State memory:	both h and c	only h	only h

Baselines



...
matching distribution
Representation extraction
...
...
Coref LSTM
DAG LSTM and embedding layers
...
...
Local
BiLSTM and embedding layers
...
...
p_1 p_2 ... p_N
...
...
q_1 q_2 ... q_M

Our model

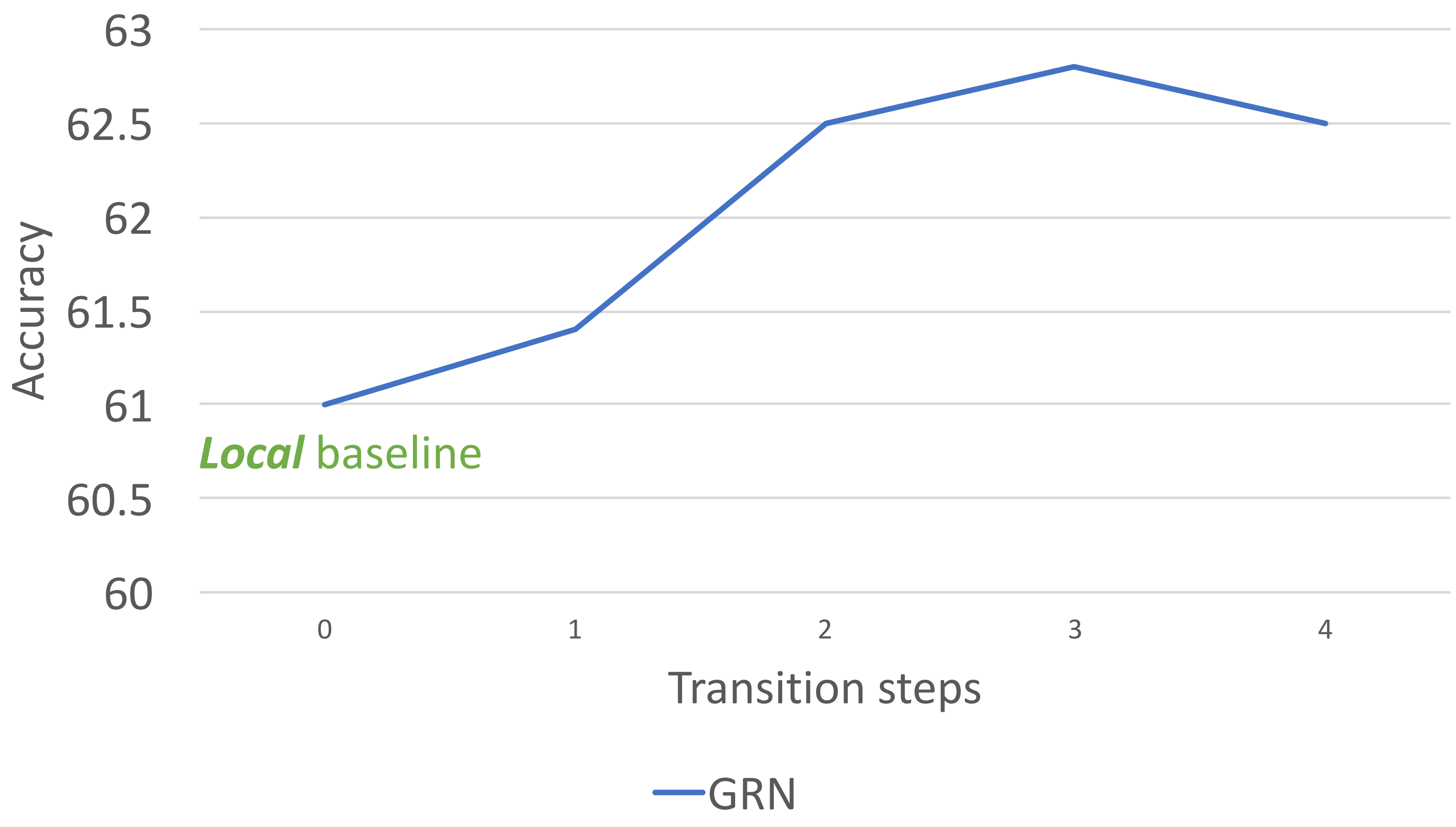


Graph step T
g_T
matching results g_T
Graph step 1
g_1
matching results g_1
Graph creation
g_0
matching results baseline
Matching results combination
Mention representations of the *Local* baseline
Question representation

Experiments

- WikiHop (http://qangaroo.cs.ucl.ac.uk/)
  - 51K instances: 44K (training), 5K (dev), 2.5K (hold-out test)
  - Each instance is: $([p_1, p_2 \dots p_L], q, C, a)$
  - Mentions are generated from automatic NER and coreference resolution, by Stanford CoreNLP

DEV experiment on message passing step (T)



Main Comparison (accuracy)

Model	Dev	Test
GA w/ GRU (Dhingra et al., 2018)	54.9	--
GA w/ Coref-GRU (Dhingra et al., 2018)	56.0	59.3
Local	61.0	--
Local-2L	61.3	--
Coref-LSTM	61.4	--
Coref-GRN	61.4	--
Fully-Connect-GRN	61.3	--
MHQA-GRN	**62.8**	**65.4**

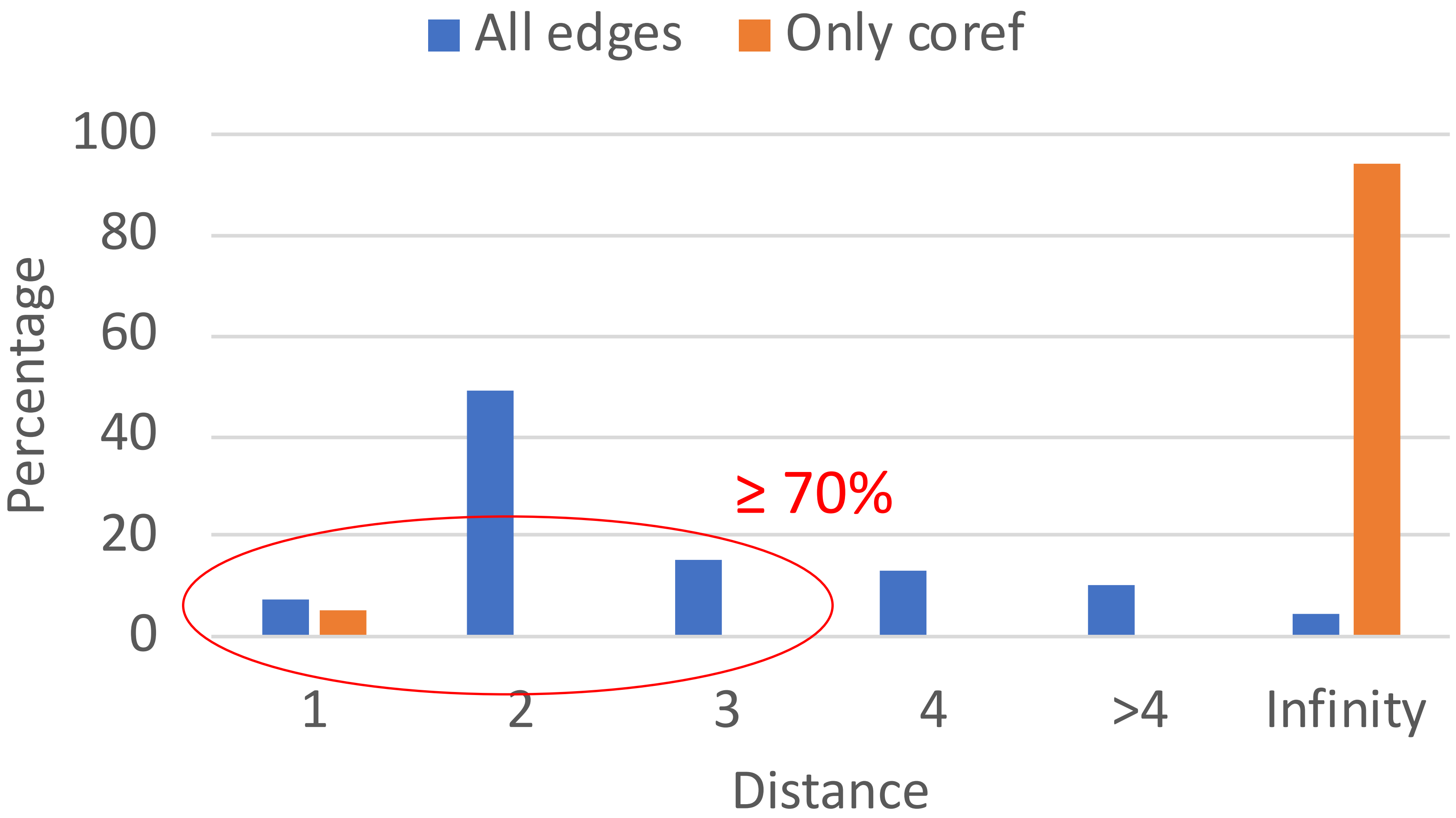
Main Comparison (accuracy)

Model	Dev		Test
GA w/ GRU (Dhingra et al., 2018)	54.9		--
GA w/ Coref-GRU (Dhingra et al., 2018)	56.0		59.3
Local	61.0		--
Local-2L	61.3	+0.4	--
Coref-LSTM	61.4		--
Coref-GRN	61.4	+1.8	--
Fully-Connect-GRN	61.3		--
MHQA-GRN	**62.8**		**65.4**

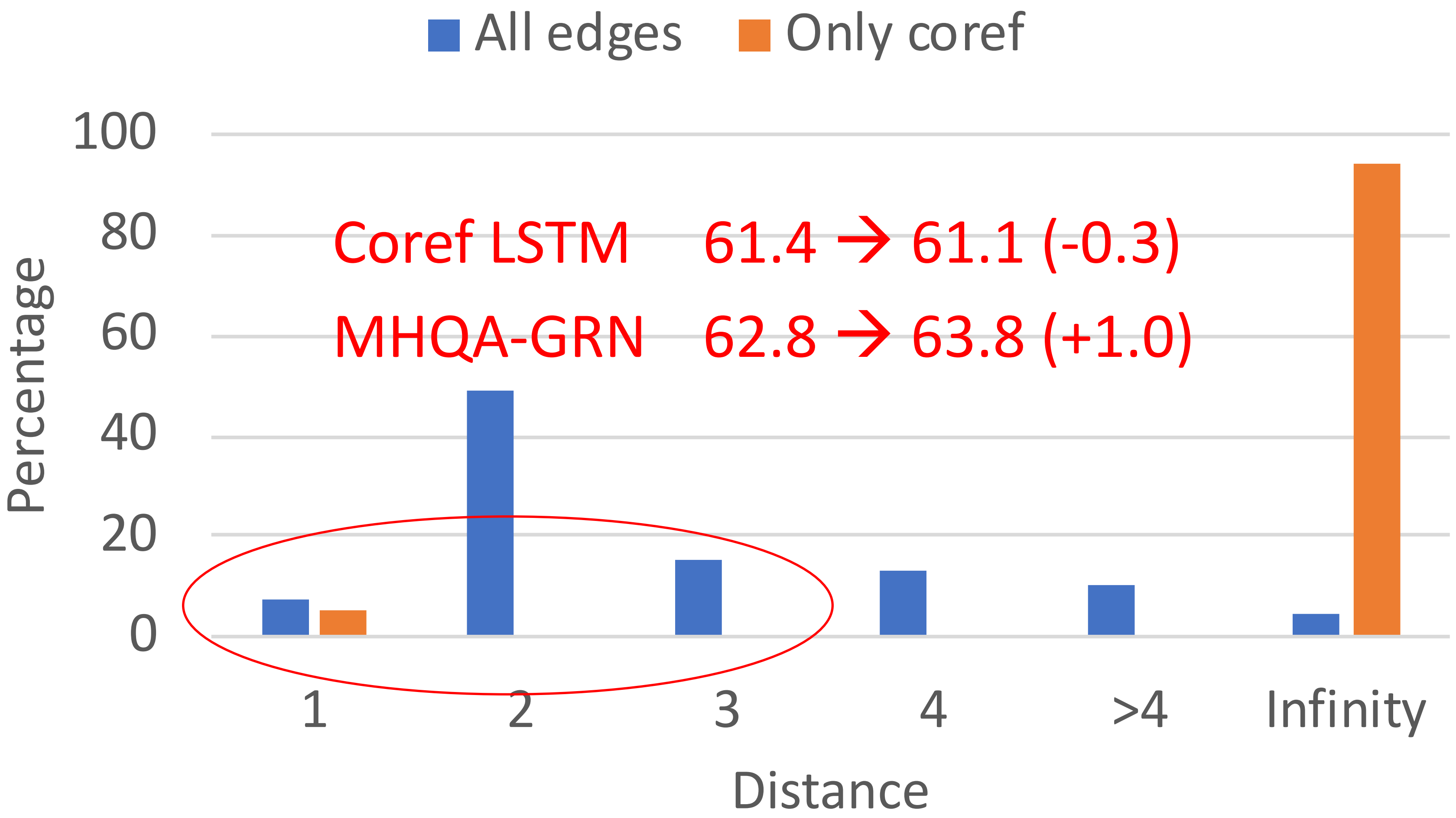
Main Comparison (accuracy)

Model	Dev	Test
GA w/ GRU (Dhingra et al., 2018)	54.9	--
GA w/ Coref-GRU (Dhingra et al., 2018)	56.0	59.3
Local	61.0	--
Local-2L	61.3	--
Coref-LSTM	61.4	--
Coref-GRN	61.4	--
Fully-Connect-GRN	61.3	--
MHQA-GRN	**62.8** +1.5	**65.4**

Distance between question and answer



Distance between question and answer



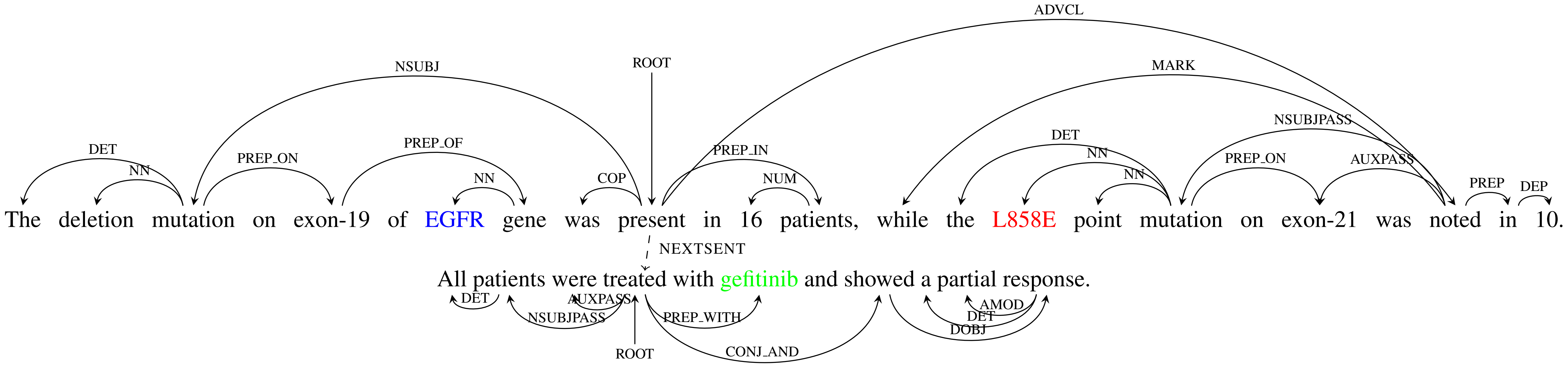
Conclusion for this work

- We introduced a new graph-based approach for evidence integration over textual knowledge.

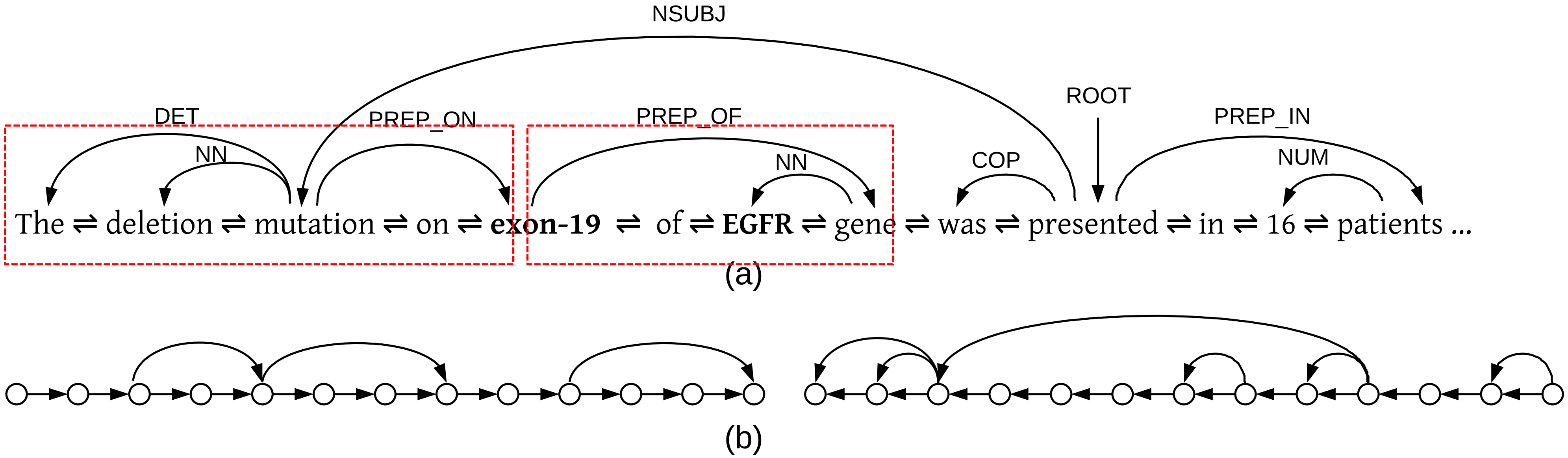
- We systematically compare with other alternatives, and we are the first to investigate a GNN on a reading comprehension task.

- Our model outperforms our strong baselines on a standard multi-hop reading comprehension dataset.

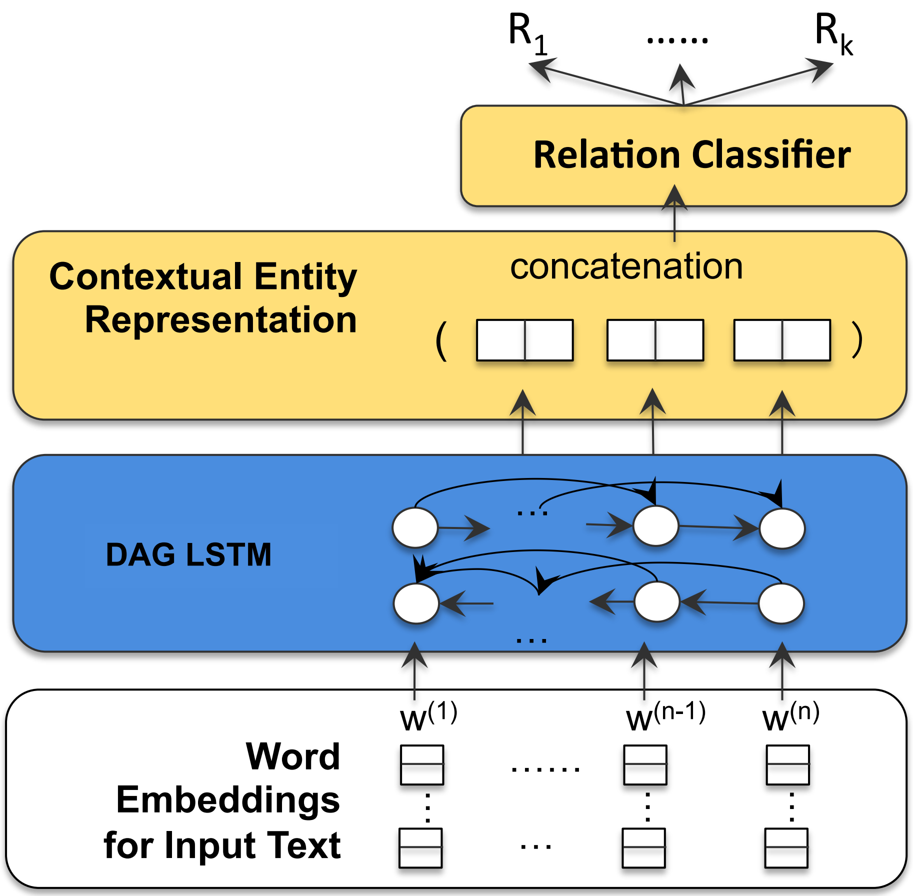
Cross-sentence *N*-ary Relation Extraction



Previous SOTA: DAG LSTM (Peng et al., 2017)



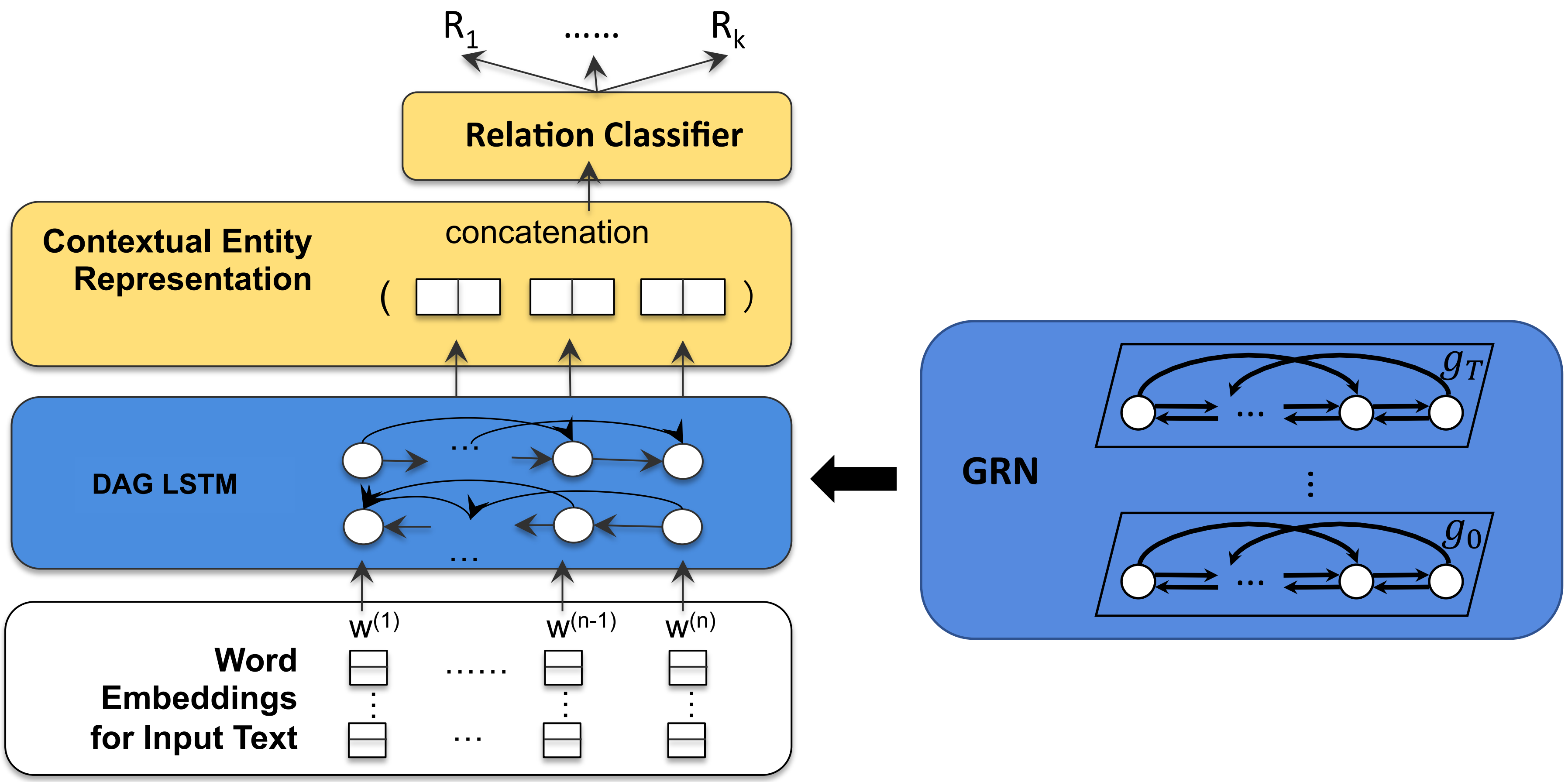
Overall framework



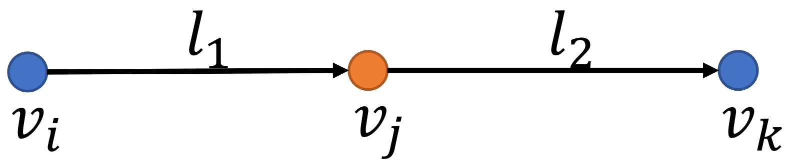
Peng et al., (2017)

Overall framework

Code available at:
https://github.com/freesunshine0316/nary-grn



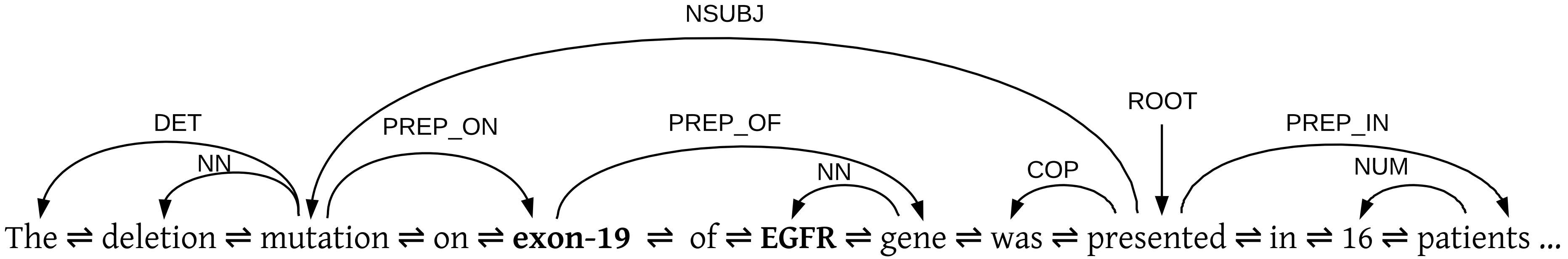
Encoding dependency graphs with GRN



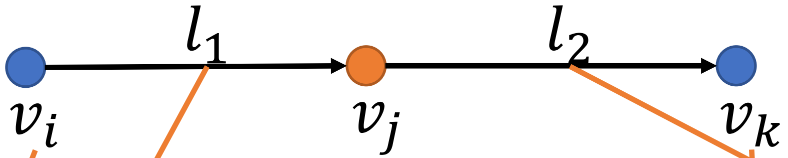
for t in $[1 \dots T]$

$m_j^t \leftarrow$ neighbors of v_j

$h_j^t, c_j^t = LSTM(m_j^t, c_j^{t-1})$

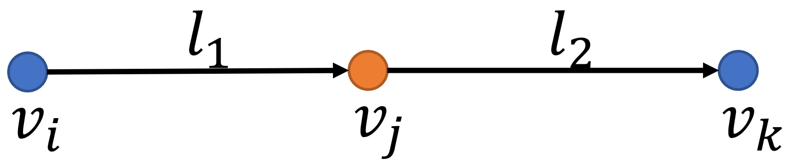


Encoding dependency graphs with GRN



$$x_{i,j}^{l_1} = W_1\left[e_{v_i}; e_{l_1}\right] + b_1 \quad x_{j,k}^{l_2} = W_2\left[e_{v_k}; e_{l_2}\right] + b_2$$

Encoding dependency graphs with GRN



$$x_{i,j}^{l_1} = W_1[e_{v_i}; e_{l_1}] + b_1 \quad x_{j,k}^{l_2} = W_2[e_{v_k}; e_{l_2}] + b_2$$

$$\phi_j^{in} = \sum_{(i,j,l)\in N_{in}(j)} x_{i,j}^{l} \quad \phi_j^{out} = \sum_{(j,k,l)\in N_{out}(j)} x_{j,k}^{l}$$

Encoding dependency graphs with GRN

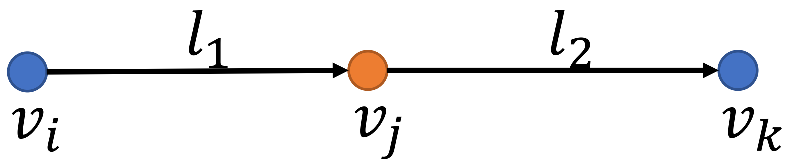
$$v_i \xrightarrow{l_1} v_j \xrightarrow{l_2} v_k$$

$$x_{i,j}^{l_1} = W_1\left[e_{v_i}; e_{l_1}\right] + b_1 \quad x_{j,k}^{l_2} = W_2\left[e_{v_k}; e_{l_2}\right] + b_2$$

$$\phi_j^{in} = \sum_{(i,j,l)\in N_{in}(j)} x_{i,j}^{l} \quad \phi_j^{out} = \sum_{(j,k,l)\in N_{out}(j)} x_{j,k}^{l}$$

$$\psi_j^{in} = \sum_{(i,j,l)\in N_{in}(j)} h_i^{t-1} \quad \psi_j^{out} = \sum_{(j,k,l)\in N_{out}(j)} h_k^{t-1}$$

Encoding dependency graphs with GRN



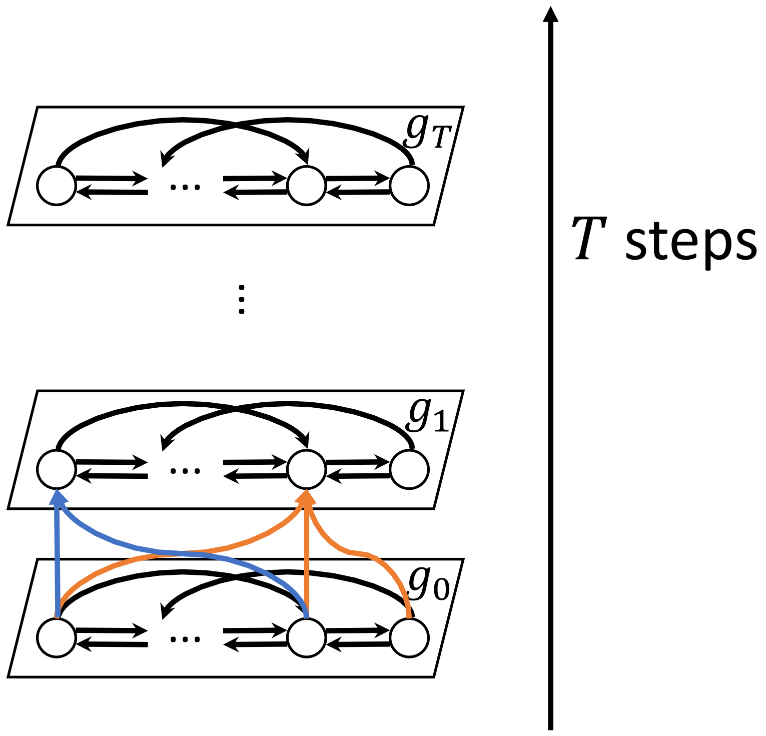
$$m_j^t = [\phi_j^{in}; \phi_j^{out}; \psi_j^{in}; \psi_j^{out}]$$

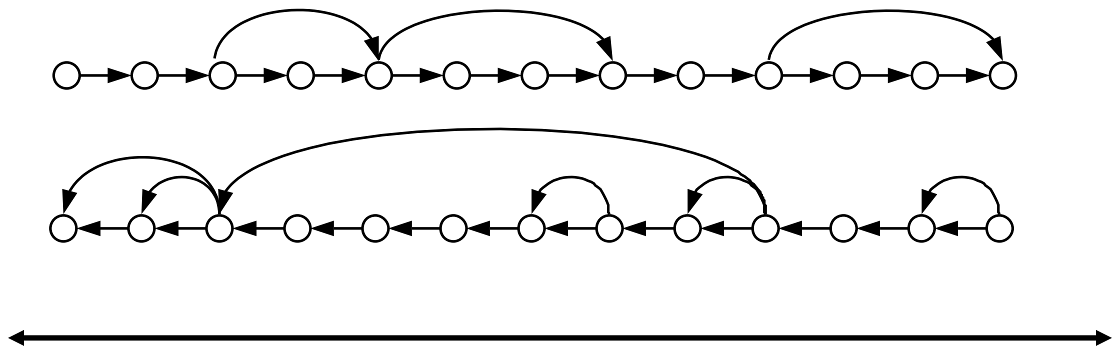
This work

$$m_j^t = \sum_{i \in N_j} h_i^{t-1}$$

Multi-hop reading comprehension

Efficiency of GRN versus DAG networks





$T << N$

Experiments

- Evaluate on the corpus by Peng et al., (2017), with annotations of dependency, discourse and entity boundaries.
  - Ternary (drug, gene, mutation): 6987 instances (Avg. length: 73.9)
  - Binary (drug, mutation): 6087 instances (Avg. length: 61.0)
- Message passing step T=5, as determined by a DEV experiment
- Evaluation (Peng et al., 2017):
  - 5-fold validation
  - Classification accuracy

Main results

Model	Precision (%)
Peng et al. (2017)	80.7
Peng et al. (2017) + Multi-task	82.0
Bidir DAG LSTM	77.3
GRN	**83.2***

Ternary

Model	Precision (%)
Peng et al. (2017)	76.7
Peng et al. (2017) + Multi-task	78.5
Bidir DAG LSTM	76.4
GRN	**83.6***

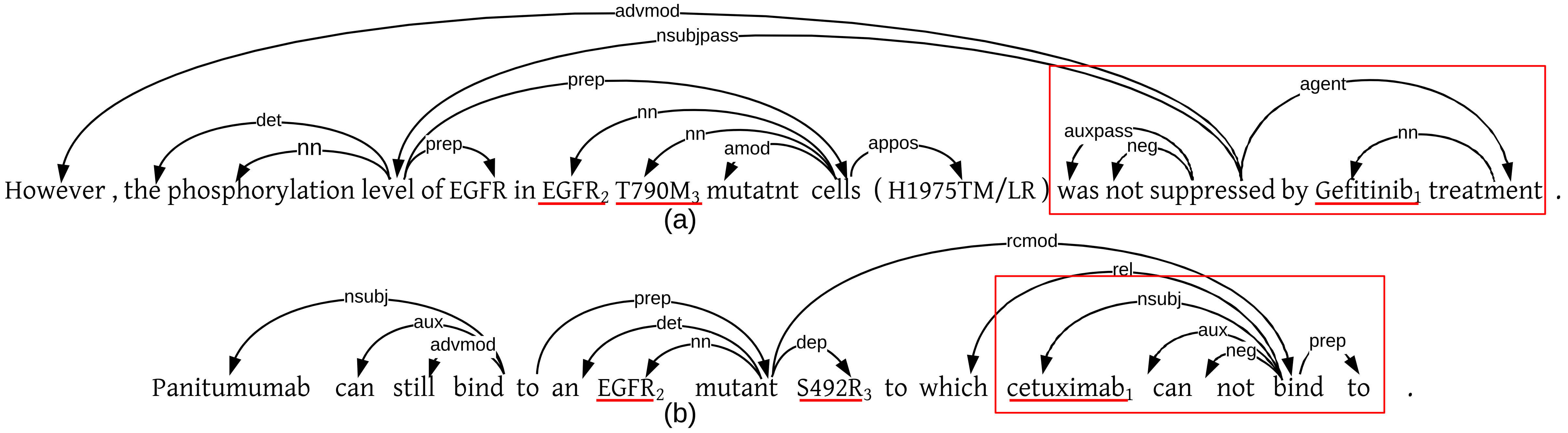
Binary

Efficiency (Ternary)

Model	Train	Decode
Bidir DAG LSTM	281s	27.3s
GRN	36.7s (7.7 times faster)	2.7s (10 times faster)

Average sentence length: 75
Message passing step: 5

Case study (Ternary)

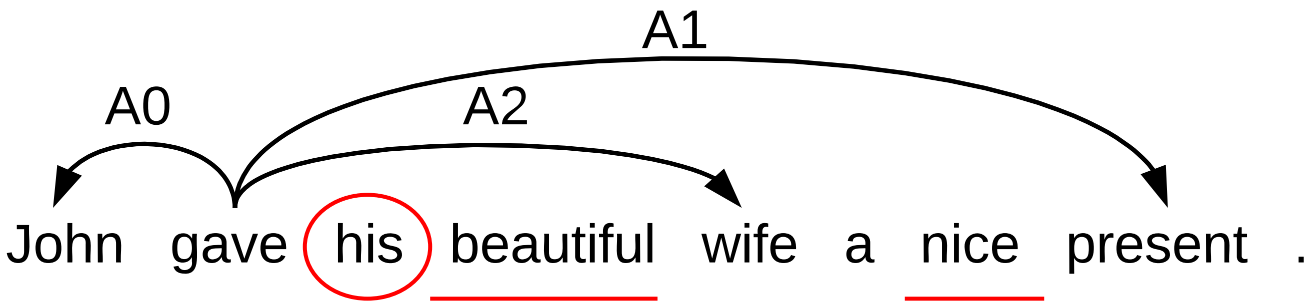


Conclusion for this work

- We studied the effectiveness of GRN for encoding dependency graphs with rich linguistic information.

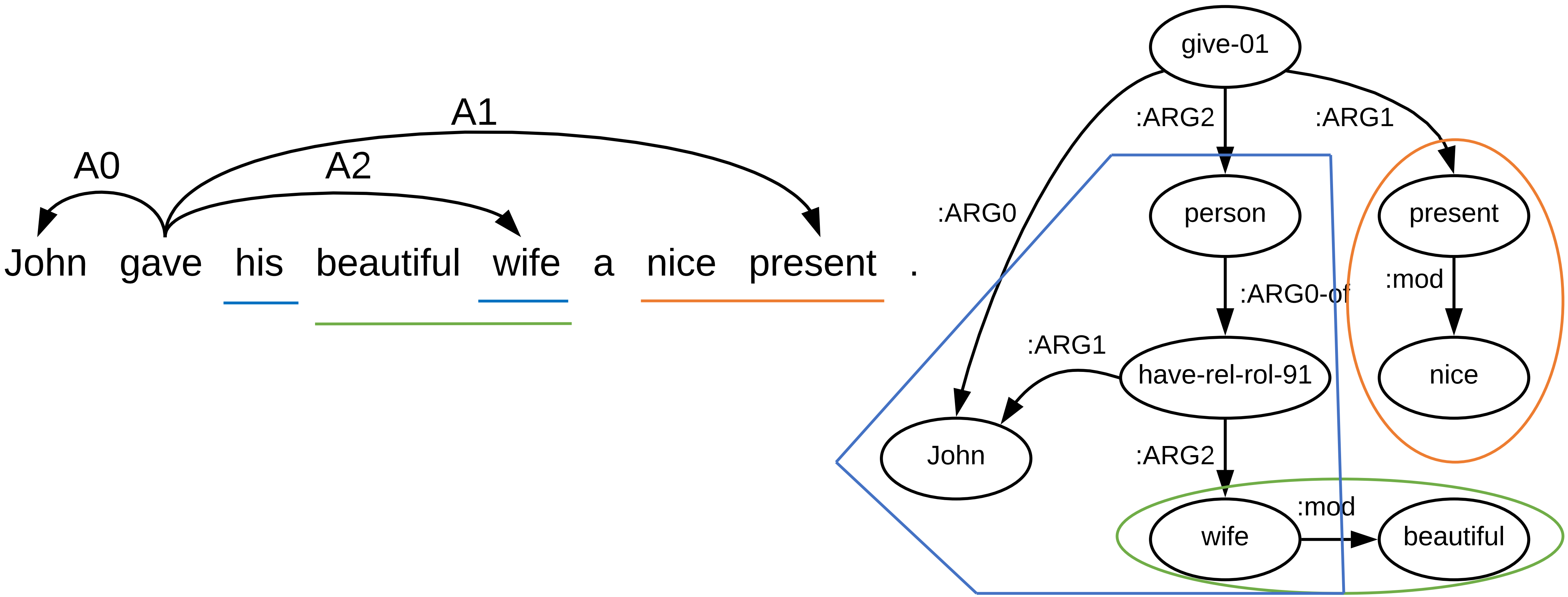
- We showed that GRN is much more effective than a DAG network by keeping the original graph structure, and it is much faster.

Semantic NMT

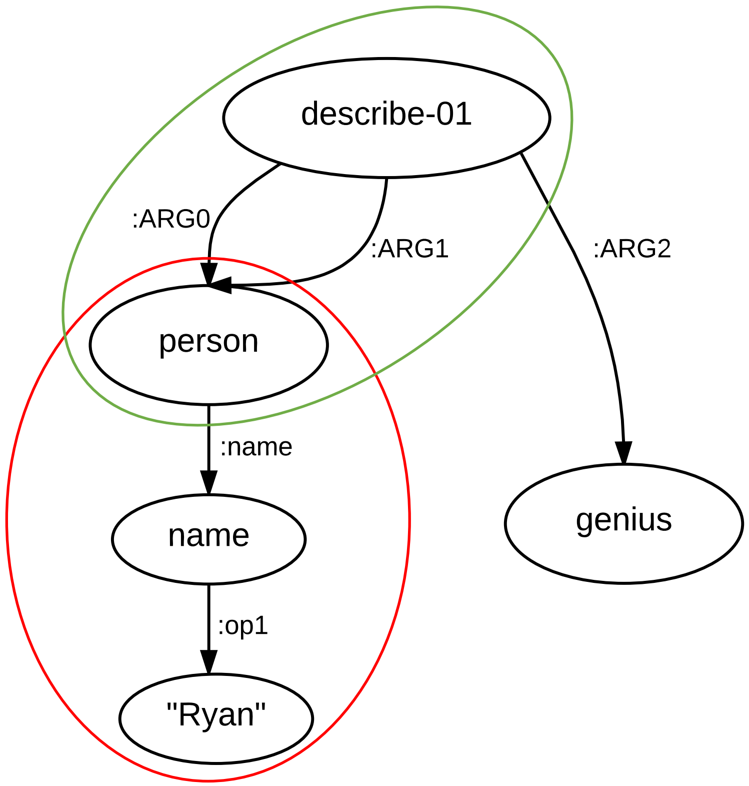


Exploiting Semantics in Neural Machine Translation with Graph Convolutional Networks. Marcheggiani et al., (NAACL 2018).

Semantic NMT using AMR

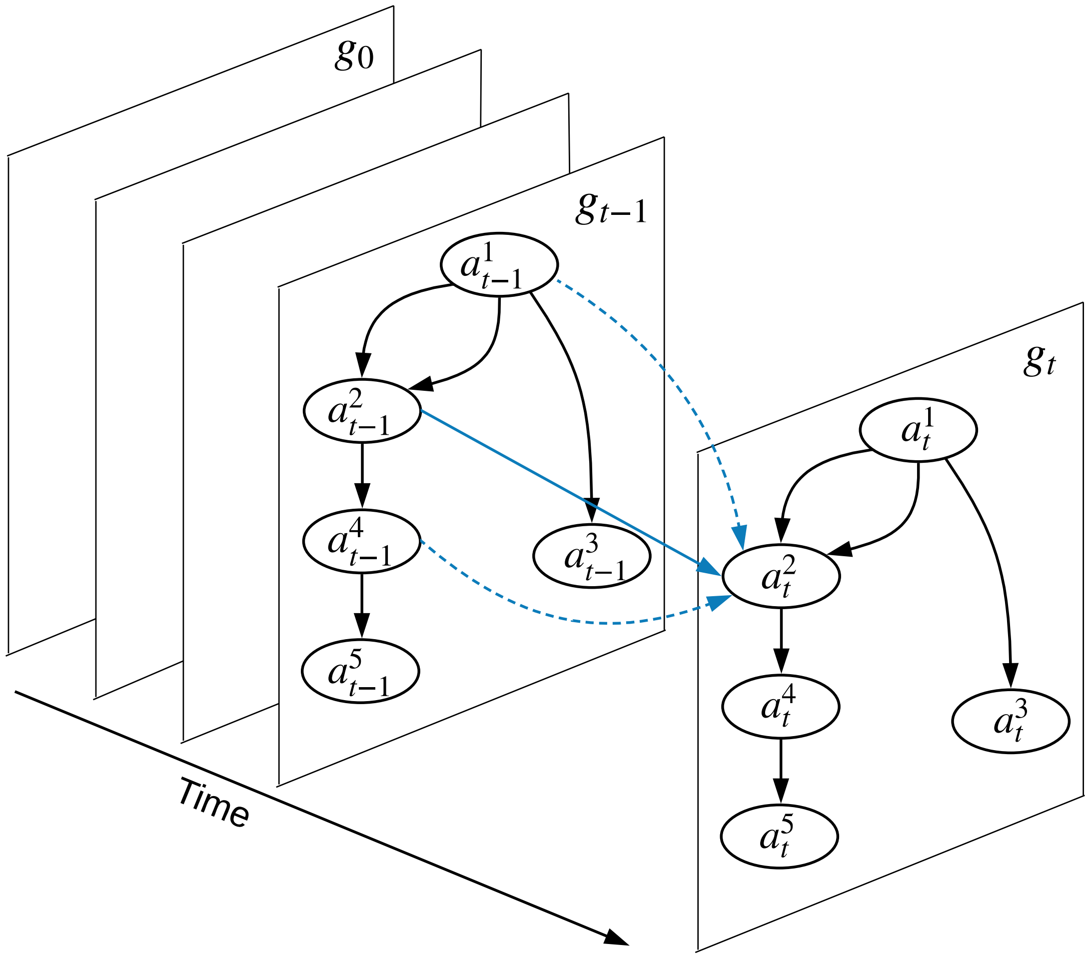


Abstract meaning representation (AMR)

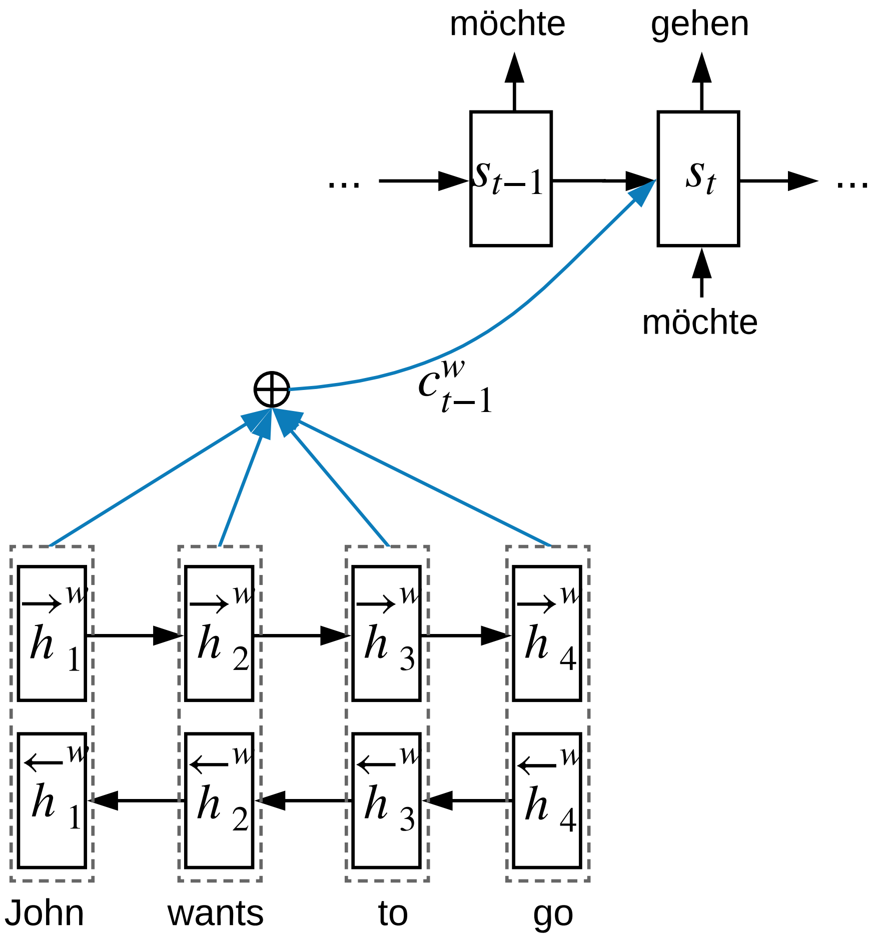


Ryan's description of himself: a genius

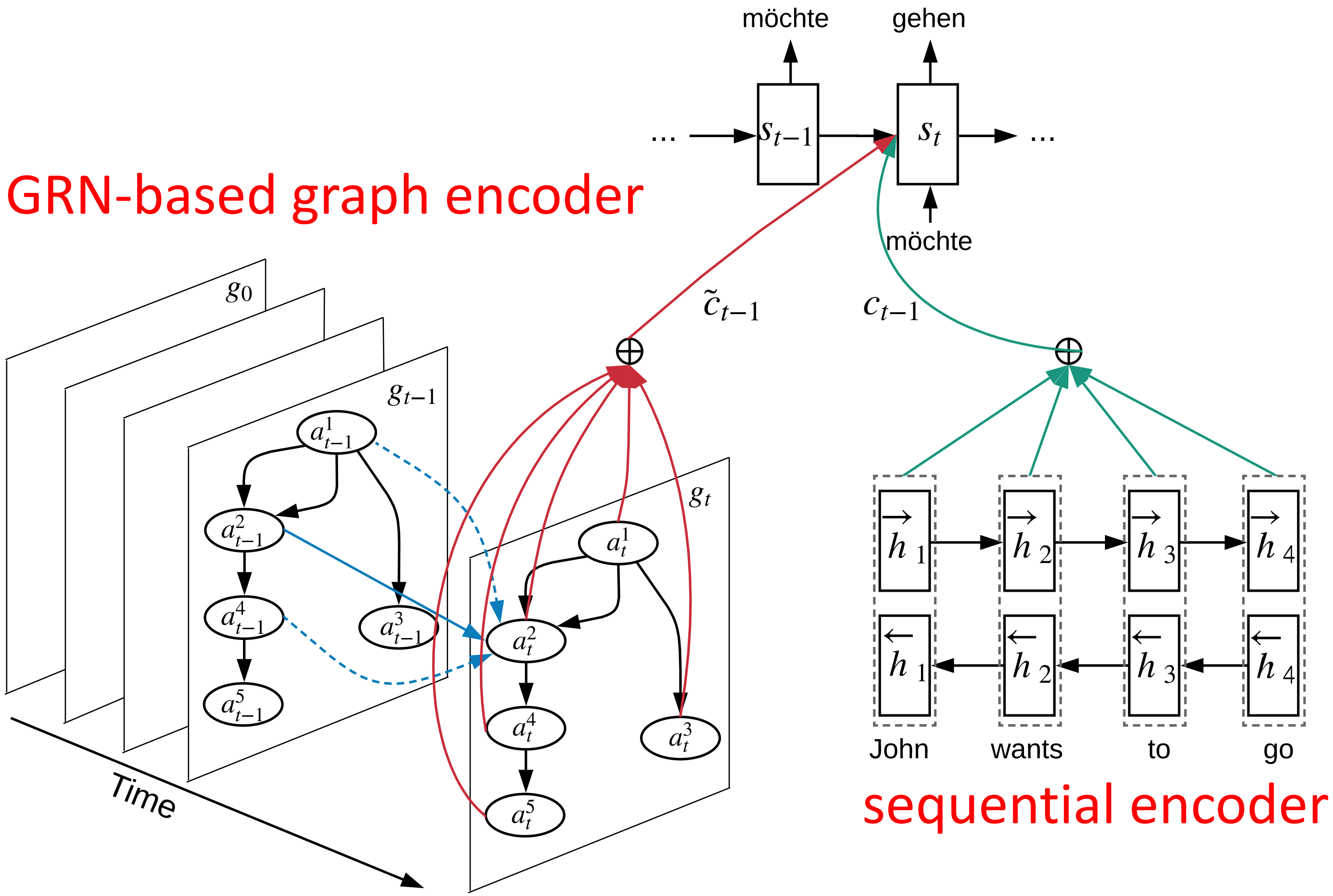
Encoding AMRs with GRN



Baseline: attention-based seq2seq



Model: Dual2seq



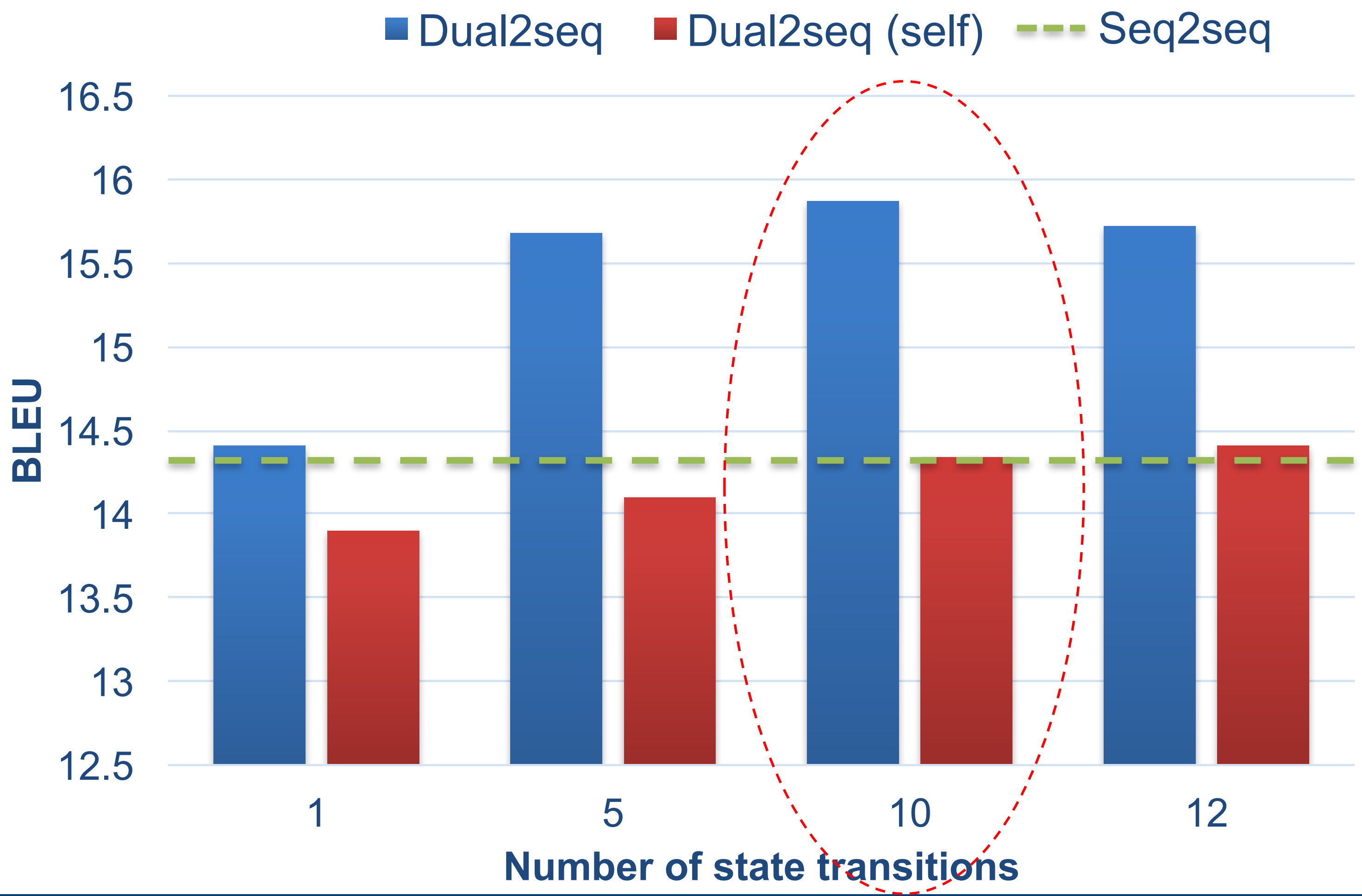
Other baselines:

- **Dual2seq-Dep**: same with Dual2seq, but GRN is for encoding dependency trees instead of AMRs
- **Dual2seq-SRL**: same with Dual2seq, but GRN is for encoding semantic roles instead of AMRs
- **Dual2seq (self)**: same with Dual2seq, but GRN is for encoding source sentences, treating it as a chain graph.
- **Dual2seq-LinAMR**: use additional sequential encoder (instead of our GRN) to encode linearized AMRs.

Experiments

- Benchmark (EN-DE):
  - Training: News commentary v11 (241K), full WMT 16 (4.5M)
  - Dev/Test: newstest2013/newstest2016
- Preprocessing:
  - Tokenization by Moses tokenizer
  - Training sentences with length ≥ 50 are filtered
  - AMRs (JAMR), dependency trees (CoreNLP), semantic roles (IBM SIRE)
- Report cased BLEU **(primary metric)**, Meteor and TER↓

Development experiments on *T*



Main results

System	NC-v11			Full WMT 16		
	BLEU(%)	TER↓	Meteor(%)	BLEU(%)	TER↓	Meteor(%)
OpenNMT-tf	15.1	0.6902	30.4	24.3	0.5567	42.3
Seq2seq	16.0	0.6695	33.8	23.7	0.5590	42.6
Marcheggiani et al. (Dep)	16.1	--	--	23.9	--	--
Marcheggiani et al. (SRL)	15.6	--	--	24.5	--	--
Marcheggiani et al. (both)	15.8	--	--	24.9	--	--
Dual2seq-LinAMR	17.3	0.6530	36.1	24.0	0.5643	42.5
Duel2seq-SRL	17.2	0.6591	36.4	23.8	0.5626	42.2
Dual2seq-Dep	17.8	0.6516	36.7	25.0	0.5538	43.3
Dual2seq	**19.2**	**0.6305**	**38.4**	**25.5**	**0.5480**	**43.8**

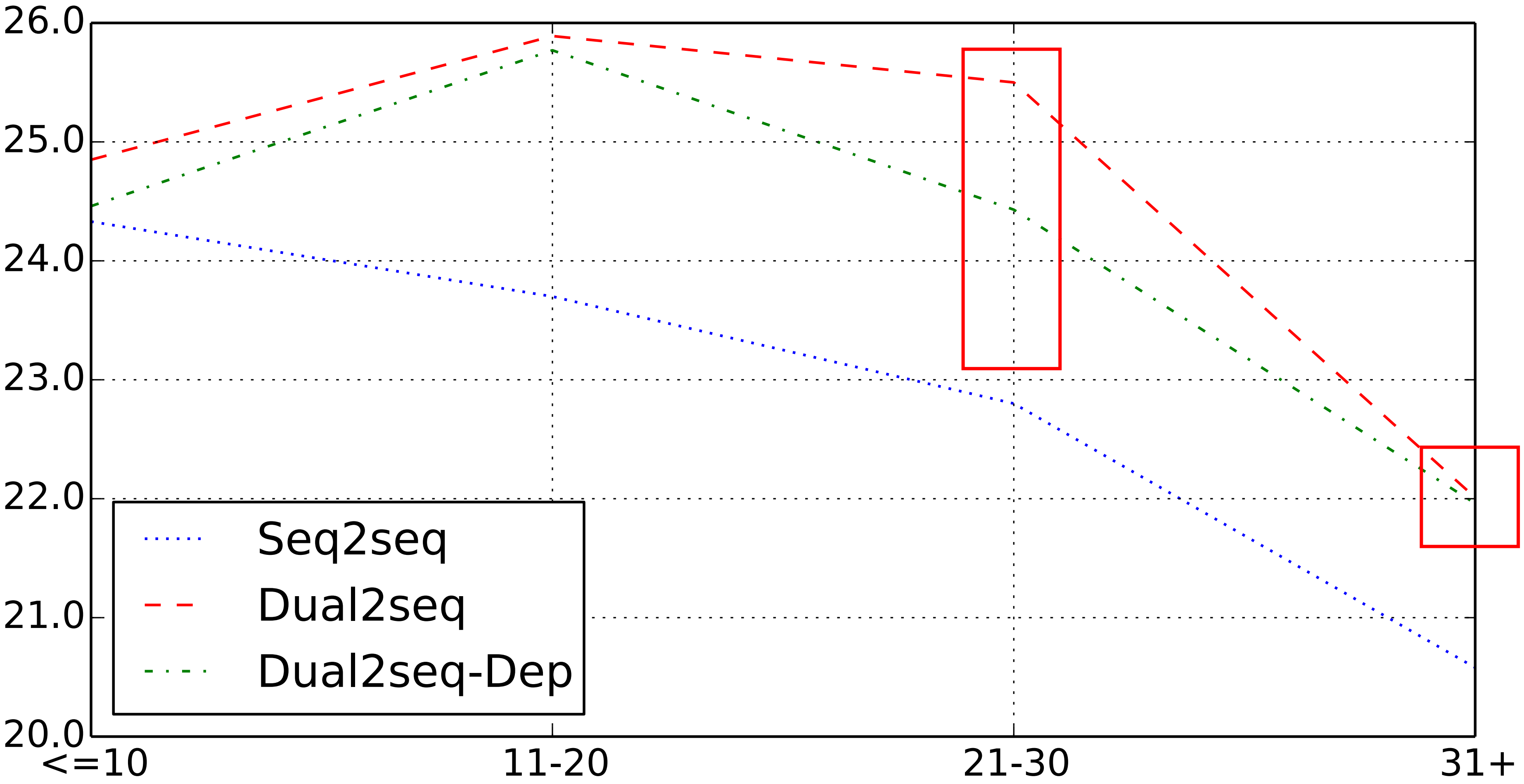
Main results

System	NC-v11			Full WMT 16		
	BLEU(%)	TER↓	Meteor(%)	BLEU(%)	TER↓	Meteor(%)
OpenNMT-tf	15.1	0.6902	30.4	24.3	0.5567	42.3
Seq2seq	16.0	0.6695	33.8	23.7	0.5590	42.6
Marcheggiani et al. (Dep)	16.1	--	--	23.9	--	--
Marcheggiani et al. (SRL)	15.6	--	--	24.5	--	--
Marcheggiani et al. (both)	15.8	--	--	24.9	--	--
Dual2seq-LinAMR	17.3	0.6530	36.1	24.0	0.5643	42.5
Duel2seq-SRL	17.2	0.6591	36.4	23.8	0.5626	42.2
Dual2seq-Dep	17.8	0.6516	36.7	25.0	0.5538	43.3
Dual2seq	**19.2** +3.2	**0.6305**	**38.4**	**25.5** +1.8	**0.5480**	**43.8**

Main results

System	NC-v11			Full WMT 16		
	BLEU(%)	TER↓	Meteor(%)	BLEU(%)	TER↓	Meteor(%)
OpenNMT-tf	15.1	0.6902	30.4	24.3	0.5567	42.3
Seq2seq	16.0	0.6695	33.8	23.7	0.5590	42.6
Marcheggiani et al. (Dep)	16.1	--	--	23.9	--	--
Marcheggiani et al. (SRL)	15.6	--	--	24.5	--	--
Marcheggiani et al. (both)	15.8	--	--	24.9	--	--
Dual2seq-LinAMR	17.3	0.6530	36.1	24.0	0.5643	42.5
Duel2seq-SRL	17.2	0.6591	36.4	23.8	0.5626	42.2
Dual2seq-Dep	17.8	0.6516	36.7	25.0	0.5538	43.3
Dual2seq	**19.2**	**0.6305**	**38.4**	**25.5**	**0.5480**	**43.8**

BLEU scores of various sentence lengths



Conclusion of this work

- We demonstrated that AMR is an effective representation for NMT and it is more useful than other common choices, such as dependency trees and semantic roles.

- GRN learns better representations for AMRs than a RNN baseline with graph linearization.

Conclusion of this talk

- We introduced our recent graph recurrent network (GRN) and its applications on several major NLP tasks

- We demonstrated that GRN successfully encodes a wide diversity of graphs and outperforms the previous SOTAs, showing that it is general and effective

Other publications: text generation

- AMR-to-text generation as a Traveling Salesman Problem. **Linfeng Song**, Yue Zhang, Xiaochang Peng, Zhiguo Wang and Daniel Gildea. In Proceedings of EMNLP 2016.
- AMR-to-text Generation with Synchronous Node Replacement Grammar. **Linfeng Song**, Xiaochang Peng, Yue Zhang, Zhiguo Wang and Daniel Gildea. In Proceedings of ACL 2017.
- Leveraging Context Information for Natural Question Generation. **Linfeng Song**, Zhiguo Wang, Wael Hamza, Yue Zhang and Daniel Gildea. In Proceedings of NAACL 2018.
- Neural Transition-based Syntactic Linearization. **Linfeng Song**, Yue Zhang and Daniel Gildea. In Proceedings of INLG 2018.

Other publications: AMR parsing

- Sequence-to-sequence Models for Cache Transition Systems. Xiaochang Peng, **Linfeng Song**, Daniel Gildea and Giorgio Satta. In Proceedings ACL 2018.

- A Synchronous Hyperedge Replacement Grammar based approach for AMR parsing. Xiaochang Peng, **Linfeng Song** and Daniel Gildea. In Proceedings of CoNLL 2015.

Other publications

- Sense Embedding for Word Sense Induction. **Linfeng Song**, Zhiguo Wang, Haitao Mi and Daniel Gildea. In Proceedings of *SEM 2016.

Thanks for listening.
Questions?